NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

PREDICTING THE ACCURACY OF UNGUIDED ARTILLERY PROJECTILES

by

Wee Yeow Lim

September 2016

Thesis Advisor:	Morris R. Driels
Second Reader:	Christopher A. Adams

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2016	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE PREDICTING THE ACCURACY OF UNGUIDED ARTILLERY PROJECTILES		5. FUNDING NUMBERS	
6. AUTHOR(S) Wee Yeow Lim			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		10. SPONSORING / MONITORING AGENCY REPORT NUMBER	

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

A method for predicting the accuracy of unguided artillery projectiles is presented in this thesis. The goal was to develop a standalone program that would estimate accuracy without the need for a large database of weapon trajectory data. The presented method uses a simplified version of the modified point mass trajectory model and error computation models to predict error metrics that are particularly useful in predicting damage effects on various types of targets using the Joint Weaponeering System (JWS). The developed program is coded in Visual Basic, and the error metrics can typically be computed in less than 30 seconds for most ranges, in the computation precision specified in this thesis.

The program was verified by comparing it against the FT 155-AM-02 firing table for the M107(HE) 155mm artillery projectile. The verification results demonstrate that the developed trajectory model closely matches the basic trajectory data to within 2% and ballistic partials to within 7% for most ranges of interest. Accuracy metrics derived from the ballistic partials generated from the developed program are within 10% of those derived from the firing table's ballistic partials for typical firing ranges. The model is able to take into account wind effects and varying levels of meteorological data staleness. The developed program is named the Indirect Fire Delivery Accuracy Program (IFDAP), and it can be used to predict accuracies for any unguided projectile given the required aerodynamic coefficients, physical properties, and error budgets.

14. SUBJECT TERMS accuracy, error, artillery, unguided, modified point mass, indirect fire			15. NUMBER OF PAGES 139
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

PREDICTING THE ACCURACY OF UNGUIDED ARTILLERY PROJECTILES

Wee Yeow Lim
Captain, Singapore Armed Forces
M.Eng., University College London, 2012

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2016

Approved by: Morris R. Driels
Thesis Advisor

Christopher A. Adams
Second Reader

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A method for predicting the accuracy of unguided artillery projectiles is presented in this thesis. The goal was to develop a standalone program that would estimate accuracy without the need for a large database of weapon trajectory data. The presented method uses a simplified version of the modified point mass trajectory model and error computation models to predict error metrics that are particularly useful in predicting damage effects on various types of targets using the Joint Weaponeering System (JWS). The developed program is coded in Visual Basic, and the error metrics can typically be computed in less than 30 seconds for most ranges, in the computation precision specified in this thesis.

The program was verified by comparing it against the FT 155-AM-02 firing table for the M107(HE) 155mm artillery projectile. The verification results demonstrate that the developed trajectory model closely matches the basic trajectory data to within 2% and ballistic partials to within 7% for most ranges of interest. Accuracy metrics derived from the ballistic partials generated from the developed program are within 10% of those derived from the firing table's ballistic partials for typical firing ranges. The model is able to take into account wind effects and varying levels of meteorological data staleness. The developed program is named the Indirect Fire Delivery Accuracy Program (IFDAP), and it can be used to predict accuracies for any unguided projectile given the required aerodynamic coefficients, physical properties, and error budgets.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

xii

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

a_o	Constant
a_1	Constant
a_x	Horizontal acceleration of the projectile
a_y	Vertical acceleration of the projectile
a_z	Lateral acceleration of the projectile
C_d	Drag coefficient
C_{m_a}	Overturning moment coefficient
C_{N_a}	Normal force coefficient
C_{l_p}	Roll damping coefficient
CEP	Circular Error Probable
CG	Center of gravity
CP	Center of pressure
d	Barrel bore diameter
D	Projectile Diameter
DOF	Degrees of Freedom
dt	Time step
FO	Forward observer
f_l	Lift force factor
f_T	Air temperature factor
f_ρ	Air density factor
F_d	Drag force
$F_{d,x}$	Drag force in x-direction
$F_{d,y}$	Drag force in y-direction
$F_{d,z}$	Drag force in z-direction
F_g	Gravitational force

F_l	Lift force
$F_{l,z}$	Lift force in z-direction
F_C	Coriolis force
F_M, F_{Mag}	Magnus force
g	Gravitational constant
GPS	Global positioning system
GUI	Graphic user interface
IFDAP	Indirect fires delivery accuracy program
I_{xx}	Axial moment of inertia
INS	Inertial navigation systems
JWAM	Joint weapon accuracy model
k	Ratio of heat capacities (C_p/C_v)
L	Twist rate of barrel in calibers
L_{ref}	Reference length
m	Mass of projectile
M	Mach number
MACS	Modular artillery charge system
MET	Meteorological (message)
MPI	Mean Point of Impact
MPMTM	Modified point mass trajectory model
p	Projectile spin rate
$\dot{p}$	Projectile acceleration
PE	Probable error
QE	Quadrant elevation

R	Individual Gas Constant
S	Reference area
t	Time
T	Air temperature
TOF	Time of flight
UAV	Unmanned aerial vehicle
V	Velocity of projectile
VE	Velocity error
v_0	Muzzle velocity
v_{imp}	Impact velocity
v_m	Mach 1 velocity (speed of sound)
v_r	Relative velocity of the projectile
v_s	Speed of sound
$v_{x,0}$	Horizontal component of the muzzle velocity
$v_{x,imp}$	Horizontal component of the velocity at impact
$v_x^{A/G}$	Relative velocity of air to projectile in x-direction
$v_x^{P/A}$	Relative velocity of projectile to air in x-direction
$v_x^{P/G}$	Relative velocity of projectile to ground in x-direction, equals to v_x
$v_{y,0}$	Vertical component of the muzzle velocity
$v_{y,imp}$	Vertical component of the velocity at Impact
v_x	Horizontal (x) component of projectile's velocity
v_y	Vertical (y) component of projectile's velocity
v_z	Lateral (z) component of projectile's velocity
w	Wind speed

X	Arbitrary independent variable used in the interpolation method
x	Distance in horizontal range
$x_{miss,TOT}$	Total horizontal miss distance
$x_{miss,MPI}$	Horizontal miss distance attributed to MPI errors
$x_{miss,Prec}$	Horizontal miss distance attributed to precision errors
Y	Arbitrary dependent variable used in the interpolation method
y	Height
z	Total drift
$z_{miss,TOT}$	Total lateral miss distance
$z_{miss,MPI}$	Lateral miss distance attributed to MPI errors
$z_{miss,Prec}$	Lateral miss distance attributed to precision errors

LIST OF GREEK SYMBOLS

a	Projectile angle of attack
a_t	Total angle of attack
a_R	Yaw of repose (Equilibrium Yaw Angle)
β	Projectile side slip angle
γ	Adiabatic index of air
$\dfrac{\delta x}{\delta V}$	Partial of range with muzzle velocity
$\dfrac{\delta x}{\delta Drag}$	Partial of range with air drag coefficient
$\dfrac{\delta x}{\delta \theta}$	Partial of range with QE
$\dfrac{\delta x}{\delta \rho}$	Partial in range with air density
$\dfrac{\delta x}{\delta T}$	Partial in range with air temperature
$\dfrac{\delta x}{\delta w}$	Partial in range with wind velocity
$\dfrac{\delta z}{\delta w}$	Partial in deflection with wind velocity
$\dfrac{\delta z}{\delta LIFT}$	Partial in deflection with lift force
$\dfrac{\delta z}{\delta a}$	Partial in deflection with azimuth angle
θ	Angle of the elevation with horizontal
θ_{imp}	Impact angle
θ_{rad}	Projectile pitch angle in radians
θ_{mils}	Projectile pitch angle in mils
θ_o	Quadrant elevation
ρ	Density of air

ϕ	Angle between x-axis and projectile velocity vector
σ	Standard deviation
$\sigma_{AIM\text{-}EL}$	Aiming in quadrant elevation standard deviation
$\sigma_{AIM\text{-}AZ}$	Aiming in azimuth standard deviation
$\sigma_{CHART\text{-}X}$	Chart accuracy standard deviation in range
$\sigma_{CHART\text{-}Z}$	Chart accuracy standard deviation in deflection
σ_{Drag}	Ballistic coefficient error budget
σ_{LIFT}	Lift error budget
$\sigma_{LOC\text{-}x}$	Location accuracy standard deviation in range
$\sigma_{LOC\text{-}z}$	Location accuracy standard deviation in deflection
$\sigma_{MPI,x}$	MPI error in range
$\sigma_{MPI,z}$	MPI error in deflection
σ_{Px}	Precision error in range
σ_{Pz}	Precision error in deflection
σ_{θ}	Quadrant elevation standard deviation
σ_{ρ}	Density standard deviation
σ_{T}	Temperature standard deviation
σ_{V}	Velocity standard deviation
σ_{w}	Wind standard deviation
σ_{x}	Horizontal range standard deviation
σ_{z}	Deflection standard deviation

ACKNOWLEDGMENTS

I thank my advisor, Professor Morris R. Driels, for his directions and advice throughout the course of the research. I greatly appreciate the numerous hours that he has spent in detailed explanation of key concepts in the field of weaponeering that allowed me to successfully develop the desired program and complete this thesis.

Thanks are also due to my loving wife for her unwavering support during the preparation of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I.   INTRODUCTION

A.   BACKGROUND

Artillery is a non-line-of-sight class of large caliber military weapons designed to propel large projectiles over long distances to cause significant damage to the adversary. Early employment of artillery focused on creating mass destruction effects, causing severe casualties, breeching fortifications, and decreasing the morale of the adversary. When timed correctly, the after effects of artillery can enable friendly forces to enter the battlefield with a significant advantage and greatly reduce casualties.

The evolution of artillery between World War I and World War II was focused on building bigger and heavier artillery, propelling larger projectiles over longer distances. Given that artillery projectiles lost most of their momentum in the initial phase of flight and that the velocity of the projectiles were limited to roughly the speed of sound [1], these weapons were inefficient, requiring huge amounts of manpower and propellants. The Paris Gun, a German long-range siege artillery gun built during World War I to bombard Paris, was 34 m long, with a bore diameter of 211 mm, and weighed 256 tons [2]. The gun was able to launch a shell 120 Km, but due to its weight and sheer size, it had to be transported on rails and required a crew of 80 men [2]. The Paris Gun was by no means a successful military weapon. The payload was small, barrel replacement was frequent, and it was highly inaccurate, making it appropriate only for city-sized targets. The labor-intensive weapon could only be fired at a maximum rate of 20 rounds / day, after which, re-boring was required [3]. Throughout its life, a mere 367 rounds were fired, of which, only 183 rounds landed within the city boundaries [3]. Its primary purpose was more as a psychological weapon than a military weapon to destroy Paris.

Modern artillery has evolved to reduce manpower requirements, and to increase automation, accuracy and mission effectiveness. In the past, the typical crew size was about 11 men to operate a M114 155mm howitzer. Today, the M109A6 Paladin Self-Propelled Howitzer needs a crew of only three, and if needed, it can execute a mission with a crew of only one. The increased automation eliminates the need to have a separate vehicle to tow the gun and for the crew to manually deploy

1

the gun. This reduces the time needed to deploy a gun as well as the susceptibility to enemy counter-fire or ambush from special forces. The weight of the guns has also been reduced to enable air transportation, increasing their capability to support a great array of military operations.

Modern-day artillery is also employed strategically to strike high-value or high-payoff targets to shape the battlefield for maneuver forces or to achieve a strategic outcome. This process is known as targeting, and its proper employment is crucial to the outcome of the overall battle.

Today's battlespace environment also presents more challenges than in the past. Greater pressure from the media and targeting of civilians decreases confidence in the military and has changed the modern battlespace into one in which collateral damage is of paramount concern. This change has expanded the role of the artillery from delivering large area effects to performing surgical precision strikes. As technology has advanced, global positioning system (GPS) and internal navigation system (INS) devices have been incorporated to guide projectiles and to provide consistent and highly accurate strikes, minimizing collateral damage. While effective, these high-tech projectiles are very costly and the ability to fire unguided projectiles accurately remains an important military requirement.

The highly complex statistical ballistic problem of accurately predicting the trajectory of an artillery projectile has been studied for decades. It is desirable for commanders on the ground to have prior knowledge of the probability of kill of a given target to aid decision making. This is especially true for targeting missions, where the predictability of the outcome is crucial. Furthermore, in an indirect fire attack, the element of surprise is the most important factor in determining the extent of casualties. Consequently, it is desirable for first and subsequent rounds to land on target. Various types of firing techniques have been developed to enhance the accuracy of artillery fire. These are briefly described in the next section.

1. Types of Artillery Fire

There are various methods of delivery an artillery projectile onto a target. These are mainly classified as unadjusted/predicted fire or adjusted fire. Unadjusted fire employs all the knowledge that can be obtained without firing a projectile to best

predict the corrections needed to bring the first projectile as near to the target as possible. These corrections typically include muzzle velocity information for that particular gun, the gun and target location and altitude, and the latest meteorological message applied to a nominal firing solution obtained from a firing table. However, the technique of unadjusted firing is unable to account for several inherent errors that cannot be known beforehand, such as the variation in propellant mass for that particular charge lot, the variation in projectile mass, gun and target location errors, and gun crew aiming errors [4].

In adjusted fire, these errors can be compensated for because one or several rounds are fired before the intended mission to observe and adjust the impact point. The technique is mainly classified into two groups [4]: observer adjusted fire and registration transfer fire. In observer adjusted fire, a forward observer (FO) is inserted to a location where he is able to observe the target. The observer records the impact points of all projectiles and transmits the corrections back to the Fire Direction Center to adjust subsequent rounds to land on target. Although this technique is more accurate, the insertion of an FO into enemy territory presents a higher risk and, by firing inaccurate projectiles into enemy territory, the element of surprise is lost. In registration transfer fire, several rounds are fired onto registration points where impact data is collected, and errors are eventually quantified into two parameters known as position line correction (PLC) and position velocity error (PVE). These parameters are used in adjusted fire techniques such as MET+VE.

Various techniques of delivery will have differing degrees of accuracy associated with them. In general, adjusted fire techniques are more accurate because they take into account errors that could not be quantified using unadjusted fire techniques.

## 2.	Errors Associated with Artillery Firing

There are three main types of errors associated with weapon delivery [4]:

1.	A fixed bias error

2.	An error that varies between occasions, called mean point of impact (MPI) error

3.	An error that varies between rounds fired, called precision error

A fixed bias error is a systematic error that is consistently present in every firing. For example, an error in the calibration of the aiming sight may cause the howitzer to be laid five milliradians off. This error may be corrected by a shift in the azimuth of the pointing direction. The MPI error is an overall quantity that accounts for several factors that vary between occasions of firing. An independent occasion in this case refers to whether a re-aiming on the target was done [4]. Some examples that would pertain to different occasions are: a displacement to a new location, traversing back to the center of arc (COA) and re-aiming, or firing at the same occasion on a different day. The precision error is a random error that accounts for the distribution in impact points from the mean point of impact in a single occasion (no re-aiming). This random error can be attributed to slight variations in the quantity of propellant between charges, or slight changes in wind speed and direction or air density.

An illustrative example of MPI and precision errors is shown in Figure 1. In this example, the gun is aiming at a target located in the origin of the coordinate frame. The gun fires four rounds consecutively without re-aiming and the impact points are as shown. An MPI can be computed from the four impact points. The distance from the target to the MPI is the MPI error, and the distance from the MPI to an impact point is the precision error.
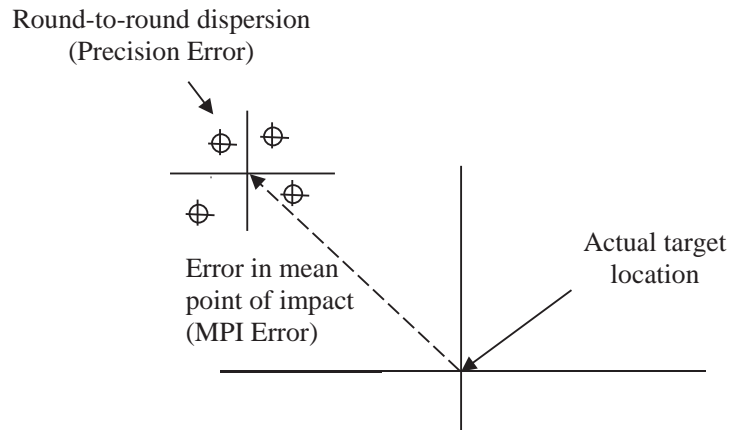
Figure 1. Precision and MPI Errors in a Single Occasion. Adapted from [4].

4

It is important to treat precision and MPI errors separately [5]. While it is possible to gather all data from decades of firing and compute a circular error probable (CEP), this number is too general and does not adequately describe the distribution of impact points associated with a unique set of conditions and technique of fire in a particular fire mission.

B. PURPOSE

The primary objective of the research is to develop trajectory models to estimate the precision of conventional ground launched, unguided ballistic projectiles using error computation models. Results of this research are intended to enable real time (<30 sec) evaluation of accuracy at a given range through the use of ballistic partials together with statistical variation of various ballistic factors, known as an error budget, determined from years of documented live firing data. Further explanation of ballistic partials and error budgets is presented in Chapter IV.

C. PROBLEM DESCRIPTION AND PROPOSED SOLUTION

There are three main components needed for a program to predict the accuracy of an unguided weapon. First, a trajectory model would take inputs such as the initial velocity of a projectile and the angle in which the projectile was fired, known as the quadrant elevation (QE), and calculate outputs such as range, time of flight, and other impact conditions. This is presented in Chapter III. Second, ballistic partials need to be calculated by running the trajectory model with perturbed input variables. Last, mathematical error models are used to quantify the accuracy of a given set of firing conditions using the relevant ballistic partials and error budgets. These are further explained in Chapter IV.

The main effort of the research is to develop external ballistic trajectory models of an artillery projectile to obtain results as close to real-world data as possible. Two trajectory models are developed with varying degrees of fidelity, taking into consideration some or all of the following factors affecting projectile flight: quadrant elevation, air density, wind, out of plane motion due to rotation of projectile, drag forces, gravitational forces, muzzle velocity, and position errors. The main projectile of interest is the Artillery 155mm M107 (High Explosive) round.

Existing high fidelity 6 Degrees of Freedom (DOF) models are able to predict the trajectory of an unguided ballistic projectile to high degree of accuracy, but these models uses complex equations that cannot be solved fast enough for strike planning purposes. This research seeks to develop approximate models using simple mathematical equations to enable a quick estimation of the accuracy.

D.   METHODOLOGY

The following highlights the approach to this research:

1.   Survey the literature on the subject. This is presented as background information and the fundamental principles of the subject of ballistics in this introduction and in Chapter II. Literature references are also used to make assumptions as discussed throughout the thesis.

2.   Develop zero drag, point mass trajectory model (using MATLAB). This is the simplest of trajectory models and serves as an introduction for non-experts.

3.   Develop modified point mass model (using MATLAB). This model is a high fidelity trajectory model that will provide good estimation of real-world data found in the artillery FT 155-AM-02 firing table. The purpose of artillery firing tables is further explained in Chapter III.

4.   Compare trajectory model outputs with data from FT 155-AM-02.

5.   Code the MATLAB model in Visual Basic and develop the QE finder model.

6.   Generate ballistic partials, the variables required to compute probable errors (PE) and MPI errors using the models.

7.   Compute accuracy metrics using error models.

E.   OUTLINE OF THESIS

Chapter I provides an introduction to artillery and briefly describes the types of artillery fire techniques and errors associated with an artillery firing. The purpose of the thesis, and the main tasks accomplished, are also briefly discussed.

Chapter II discusses the ballistics involved. This chapter mainly describes the phenomenon and forces encountered by a projectile through the three phases of internal, external, and terminal ballistics, along with the describing equations that

quantify these forces. This chapter also provides the reader with a sufficient degree of understanding of projectile ballistics in order to understand the trajectory models.

Chapter III presents the MATLAB trajectory models and the kinematic and mathematical equations used to derive the presented results. A comparison with FT 155-AM-02 is also presented.

Chapter IV presents the two error models, namely precision and MPI error models, used to derive accuracy statistics.

Chapter V presents the implementation computation flow using the models in Chapters III and IV coded in MATLAB with an example case study and in Microsoft Excel Visual Basic for operational usage.

Chapter VI validates the implemented models over a comprehensive range of data from FT 155-AM-02.

Chapter VII discusses various key observations made during the work of this thesis.

Chapter VIII concludes the thesis with some important observations.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BALLISTICS

Ballistics, in a simple sense, is the engineering study of the motion of a projectile to achieve a desired performance. Ballistics is generally divided into three phases: internal ballistics, which studies the propulsion; external ballistics, which studies the flight; and terminal ballistics, which studies the detonation effects. This thesis focuses on quantifying the various factors in internal and external ballistics to provide accurate estimations on the accuracy of an artillery firing.

A. INTERNAL BALLISTICS

Internal ballistics deals with the events that take place when the projectile is in the barrel of the gun. In the case of a conventional artillery gun firing a non-bleeding projectile, all the energy that is required for the projectile to be propelled through the air is imparted in the barrel by igniting solid propellants known as charges. In modern artillery platforms, the charges are molded in blocks called a Modular Artillery Charge System (MACS), which provides flexibility to the user in terms of varying muzzle velocity according to the number of charges put in place. The ignition of the charges produces hot and high pressure gases over a short period of time, which pushes the projectile through the barrel, launching it into the air.

As the projectile navigates through the barrel, the wall of the barrel serve four important purposes: 1) it provides directional guidance to the projectile; 2) it aligns the axis of the projectile in a particular direction; 3) it causes the projectile to spin, which provides stability to during the flight; and 4) it helps to absorb the recoil [1]. The spinning of the projectile is caused by engravings, known as rifling, along the walls of the barrel. The spinning motion generates stability for the projectile, which is discussed later in Chapter II-B. Rifling is analogous to a screw thread, but spread further apart. If you follow a rifling down the barrel, eventually it will make one full circle. The length of the barrel required to make one full circle, is known as twist, which is usually measured in terms of calibers, the unit length for the diameter of the projectile. The smaller the twist length, the faster the spin of the projectile. The engravings of the rifling form lands and grooves, which are parts of the wall that are protruding out and cut in. The projectile has a ring called a rotating band near the rear, which is usually made of a soft metal such as a special copper alloy. As the projectile

travels through the barrel, the lands cut the rotating band, imparting spin onto the projectile. The friction produced causes barrel wear over time, which reduces the pressure built up in the chamber and reduces muzzle velocity, thus affecting the accuracy. Heat produced from repeated firings also expands the barrel, producing the same undesirable effect.

In an ideal scenario, a firing should not produce any movement in the barrel. However, due to the high forces involved, small movements are unavoidable. For instance, a long barrel gun may sag a little, but during firing, the internal pressure straightens the drooping gun causing a slight increase in the elevation [1]. This phenomenon is called a whip. The large recoil force may also cause the platform to move, called a hop. These movements result in the change in the initial direction of laying, which is called a jump. These motions are undesirable and affect the accuracy of the projectile.

B.   EXTERNAL BALLISTICS

External ballistics deals with events that take place from the moment when the projectile leaves the barrel to the moment just before detonation. This may or may not be at impact, as further discussed in Chapter II-C2 on various fuze types. In this section, the various forces and phenomena that a projectile encounters and the equations that quantify them are presented. This is accompanied by the kinematic equations used to compute the trajectory, which are presented in Chapter III.

1.   Drag

Suppose that a projectile is fired in a vacuum. The only force that affects the trajectory of the projectile is gravity, which acts approximately equally throughout the flight. This results in a trajectory that would be symmetrical, and the maximum ordinate would be at half the range. Air resistance reduces the range of the projectile and causes a higher angle of impact than the angle of gun elevation. As a result, the maximum ordinate (maximum altitude) of the projectile would be somewhere beyond half range, and the trajectory would not be symmetrical about the apex. Figure 2 shows the differences in trajectory for a projectile experiencing zero drag and with drag.

10
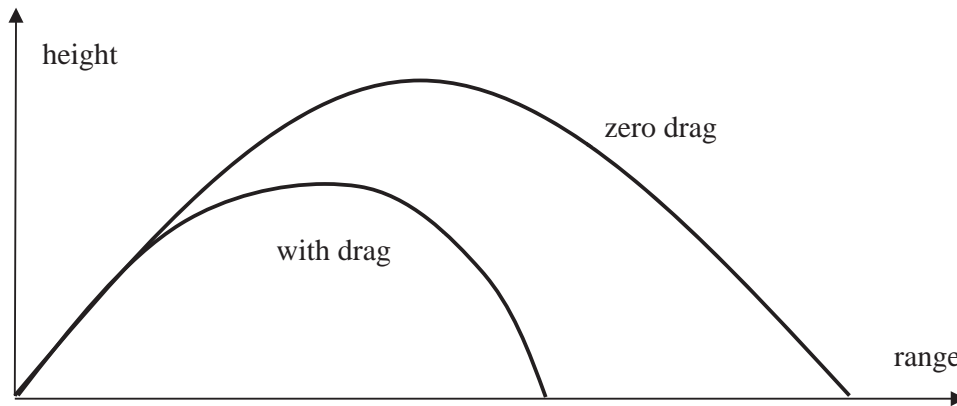
Figure 2. Effect of Drag on the Trajectory of a Projectile. Source: [4].

Air resistance exists in various forms. First, as the projectile travels through the air, it pushes aside and compresses the air in front of it. This compressed air tends to oppose the projectile, creating a forebody drag, which increases with velocity. Second, at the rear, a momentarily low pressure region is generated behind the base as air displaced by the body of the projectile creates a partial vacuum. This is known as base drag, which tends to draw the projectile toward the partial vacuum. Finally, air that flows around the body of the projectile generates skin friction, further reducing the velocity of the projectile. These three forms of air resistance vary differently with the velocity of the projectile. Forebody drag increases linearly up to the speed of sound, beyond which, forebody drag increases substantially with increasing velocity. Base and spin drag tend to level off as the velocity of the projectile approaches Mach 1 (Figure 3).
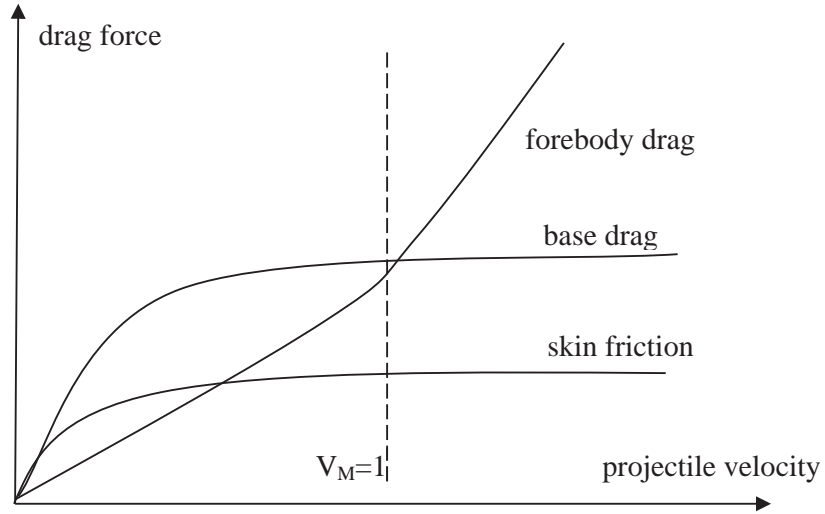
Figure 3. Variations of Various Forms of Resistance with Projectile
Velocity. Source: [4].

The total drag force, F_d, acts in the direction directly opposite to the velocity vector (Figure 4) and is a function of projectile velocity relative to air and the density of air, described by the following equation:

$$F_d = \frac{1}{2} S C_d \rho V^2$$

(1)

where S is the reference surface area,

C_d is drag force coefficient,

ρ is the density of air, and

V is the projectile total velocity
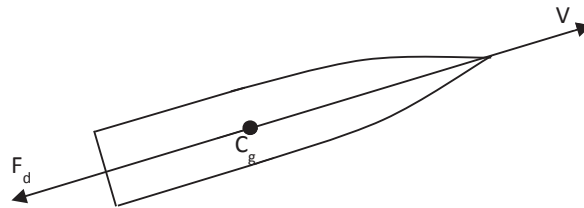
Figure 4. Direction of Total Drag Force Acting on a Projectile in Flight

2. Stability

The stability of a projectile is determined by the position of its center of gravity (CG) relative to its center of pressure (CP). The CP is an imaginary point along the axis of the projectile such that the sum of all aerodynamic moments is zero, whereas the CG is a point along the same axis such that the moments caused by its own weight sum to zero. For a projectile to be stable, the CP must be behind the CG (Figure 5). Otherwise, the projectile will tumble forward.

center of gravity

center of pressure

center of gravity

center of pressure
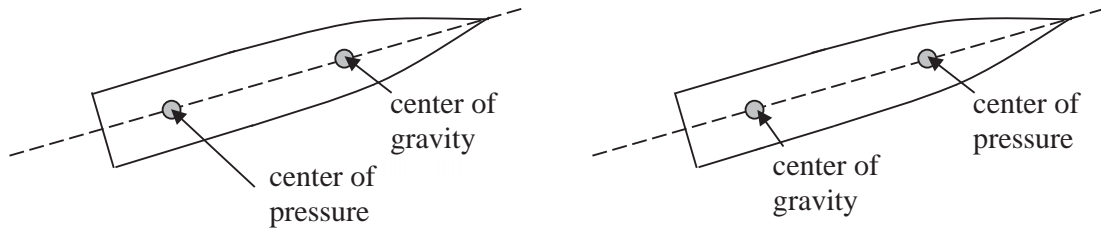
Figure 5.   Stable (left) and Unstable (right) Projectiles. Source: [4].

The ogive-shaped nose of an artillery projectile that is critical for reducing forebody drag will always result in the location of the CG behind the CP, hence causing an unstable flight. Therefore, some means of stabilization is necessary. In general, a projectile may be stabilized by fins or by spinning the projectile. Fins provide additional aerodynamic moments at the aft of the projectile such that the resultant CP is shifted to the rear. These fins can also steer the projectile during flight to the trajectory such that the projectile falls accurately on target. While effective, fins are delicate components. The ignition of charges in the barrel chamber is explosive in nature and imparts setback accelerations in excess of 15,000Gs [1]. Fins designed to sustain such high forces can be expensive to manufacture. Since artillery howitzers are usually employed for mass effects purposes where tens of projectiles are fired in a single mission, spin stabilization is the more economical option.

3. Lift (Out of Plane Motion)

Ideally, a projectile fired at an elevation from the ground without the presence of wind will only have velocity components in the vertical plane. However, if the projectile is spinning, an out of plane motion will occur due to gyroscopic drift.

a. *Gyroscopic Effects of Spin*

A spinning projectile is governed by the gyroscopic effect, which tends to maintain the orientation of the axis of spin. This effect is commonly observed in spinning tops, which stay upright because the spin produces a resultant force acting along the axis of spin. For an upright spinning top, this force counteracts gravity and prevents the spinning top from falling over [6]. As the spinning top slows down, the magnitude of this force reduces until it is overcome by gravity and topples over (Figure 6).

F_l

Angular Momentum

F_g
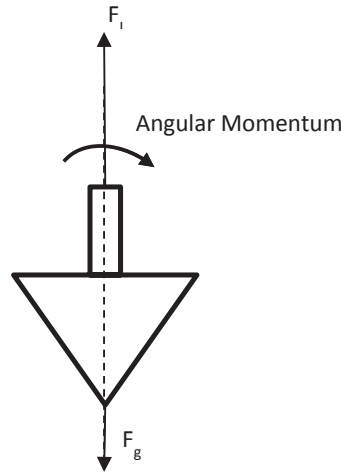
Figure 6. Direction of Lift and Gravitational Forces in a Spinning Top

A non-spinning projectile has a resultant overturning moment that causes the projectile to tumble forward. Similar to the spinning top, the spinning projectile produces a force that counteracts this overturning moment and maintains the orientation of the spinning axis (Figure 7).

F_M

F_M F_L Angular Momentum
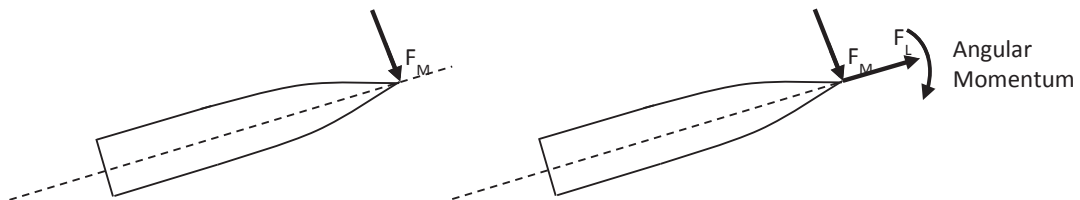
Figure 7. Direction of Overturning Moment and Lift Forces in a Spinning Projectile.

14

When a disturbance is applied to a spinning top, a torque will be generated along the axis that the force is trying to rotate the object. This axis, and the direction of the torque, would be 90 degrees from the direction of the force. The torque causes the axis of spin to move in its direction, resulting in the precession of the top (Figure 8).
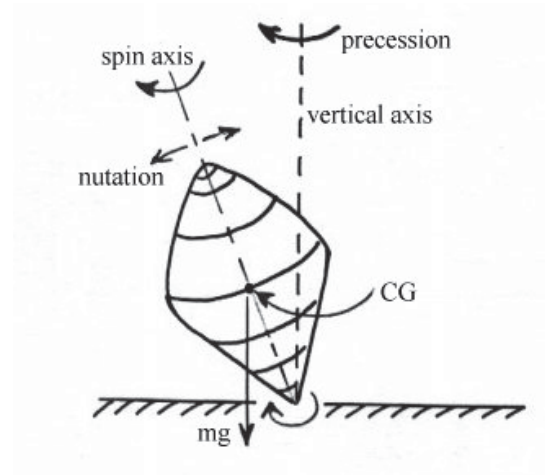
Figure 8. Precession of a Spinning Top. Source: [7].

In the case of a spinning Artillery projectile, the overturning moment in the vertical direction causes a torque to be generated along the lateral direction (Figure 9). This causes another torque to be generated in the vertical direction, resulting in precession of the projectile.
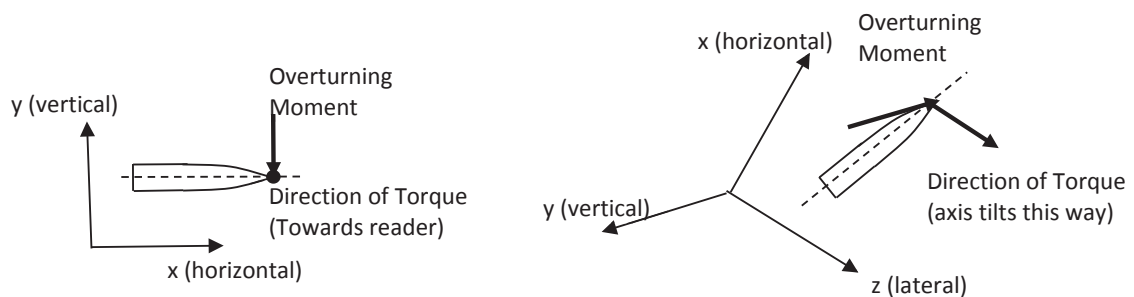
Figure 9. Initial Overturning Moment and Direction of Torque in a Spinning Projectile

As the spin rate slows down due to skin friction, the precession dampens out. This is known as nutation (Figure 10), and the resulting trajectory would be a precession about its trajectory instead of a straight line (Figure 11).
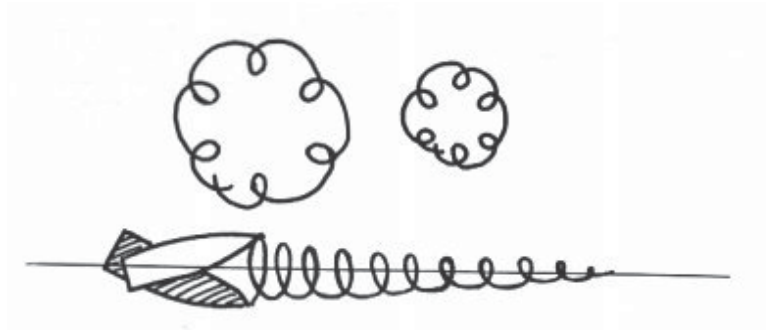
Figure 10. Nutation of an Artillery Projectile. Source: [7].
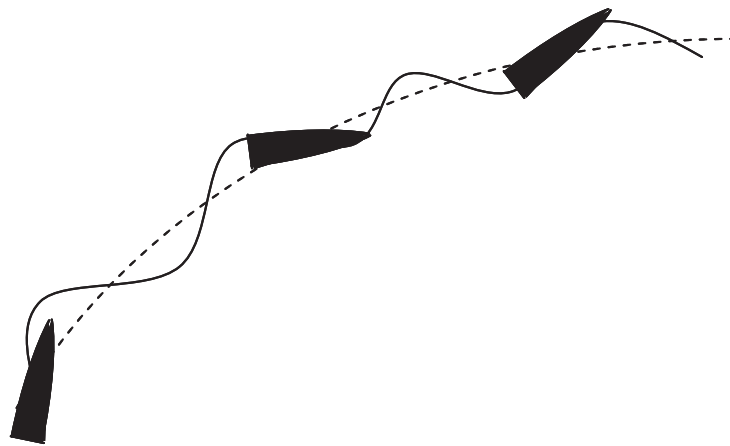
Figure 11. Precession of an Artillery Projectile along its Trajectory.
Source: [7].

b.   *Spin Rate*

Generally, the rate of spin equates to the magnitude of stability of the projectile. If the rate of spin is too high, the projectile can be over stabilized and land on its base instead of its nose (Figure 12). If the rate of spin is too low, the projectile will be insufficiently stable and continue to tumble over. The design of an ideal spin rate balances the required stability for flight while allowing for pitching motion such

16

that it may impact fuze first. The angular velocity about the projectile's axis is denoted as p. The initial spin rate p_0, can be evaluated from the following equation:

$$p_0 = \frac{2\pi v_0}{Ld}$$

(2)

where L is the twist rate, in calibers, as explained in Chapter II-A.
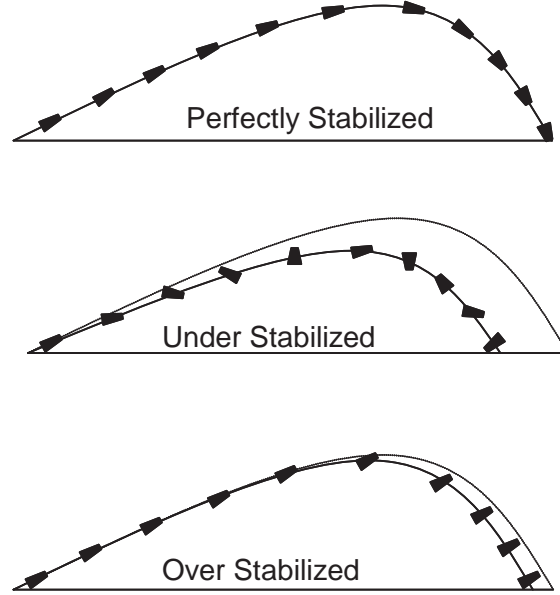
Figure 12. Behavior of Projectile at Various Stability Levels. Source: [7].

c.   *Spin Damping Moment*

As the projectile travels through the air, skin friction and moments of inertia reduce the spin rate over time. The deceleration in spin rate can be calculated from the spin damping moment equation as follows [8]:

$$M_p = \frac{1}{2}\rho V^2 Sd\left(\frac{pd}{V}\right)C_{lp}$$

(3)

$$\dot{p} = \frac{M_p}{I_{xx}}$$

(4)

where M_p is the spin damping moment.

Since the aerodynamic coefficient C_{lp} is always negative, the change in angular velocity will always be negative.

d.   *Yaw of Repose*

A right-hand spinning projectile always starts precessing from the right. As the spin rate decreases during the precession, the magnitude of the torque that displaces the projectile's axis from the direction of fire is larger than the torque that returns it back. This results in a net yaw angle known as the yaw of repose. The 3D yaw of repose has two components: an angle of attack, and a side slip angle (Figure 13 and Figure 14). The yaw of repose is not constant throughout the flight. It increases initially as the torque that initiates the precession is larger than the overturning moment and decreases when the torque drops below the magnitude of the overturning moment that destabilizes the projectile with decreasing spin rate. The yaw of repose has three effects on the trajectory of the projectile: 1) it produces a lateral force that results in a the projectile drifting right (for a right-hand spinning projectile); 2) it produces some lift, which increases range; and 3) it increases the total drag due to an additional yaw drag component. The additional lift is of a very small magnitude, and the yaw drag component is a second order term; thus, they will be omitted for the purposes of this thesis. The magnitude of the yaw of repose, α_R, can be evaluated from [9]:

$$\overline{a_R} = \frac{8 p I_{xx}}{\pi \rho D^2 L_{ref} C_{m_a} V^4} (\vec{g} \times \vec{V})$$

(5)

It is usual to define the side slip angle as the projection of the yaw of repose on the horizontal plane [9]:

$$\beta = \frac{8 p I_{xx} g \cos \theta}{\pi \rho D^2 L_{ref} C_{m_a} V^3}$$

(6)

18
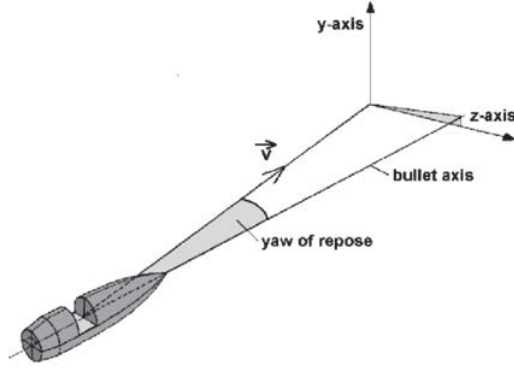
Figure 13. Gyroscopic Drift due to Yaw of Repose. Source: [10].
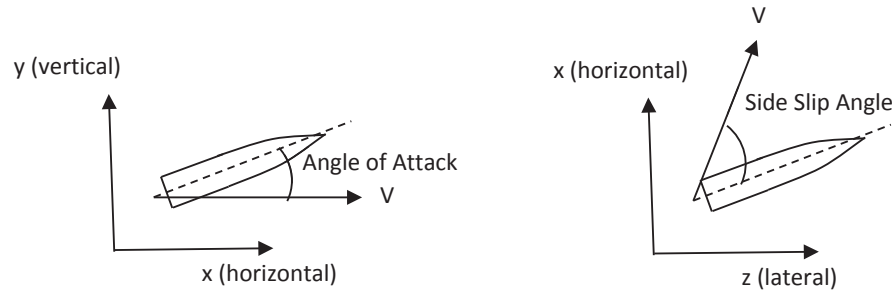
Figure 14. Side (left) and Plan (right) View of Yaw of Repose

e. *Lateral Lift (Drift)*

When a projectile's velocity vector is not aligned with its axis of symmetry, there exists an angle of attack, a, and/or side slip angle, β. This produces a lateral lift to the right, for a right-hand spinning projectile, which results in the out of plane motion. The lateral force, $F_{l,z}$, due to yaw of repose is estimated using the following equation [8]:

$$F_{l,z} = \frac{1}{2}\rho V^2 S C_{N_a} \sin(a_t)$$

(7)

where C_{N_a} is the normal force coefficient, and

a_t is the total angle of attack given by [8]:

$$\sin(a_t) = \sqrt{(\sin a \cos \beta)^2 + \sin^2 \beta}$$

(8)

19

In the trajectory model developed in this thesis, it is assumed that the angle of attack is negligible. This assumption is justified later. Hence, Equation 8 is reduced to:

$$\sin(a_t) = \sin(\beta)$$

(9)

and Equation 7 becomes:

$$F_{l_z} = \frac{1}{2}\rho V^2 S C_{N_a} \sin\beta$$

(10)

Note that if the normal force coefficient is used, the lift force is normalized to the orientation of the projectile.

4.   Coriolis Effect

While the angular velocity of the Earth is the same at all latitudes, the surface velocity is different due to the difference in distance from the axis of spin. At the equator, distance is furthest from the spinning axis and therefore has a highest ground velocity as compared to other latitudes. This phenomenon results in a trajectory that would not be straight if the projectile was fired into either hemisphere from the equator. Since the tangential surface velocity at the equator is the highest, if the projectile is fired toward the northern hemisphere from the equator, it would carry with it a tangential surface velocity higher than the northern hemisphere. As a result, the projectile would move to the right faster than the surface velocity of the earth at the northern hemisphere, resulting in a drift to the right of the intended path. This would be the exact opposite if the projectile is fired into the southern hemisphere. This effect is depicted in Figure 15. Although it affects the trajectory of the projectile, for a given set of firing conditions, the Coriolis effect does not vary between occasions, and therefore, it does not affect accuracy. It is also the objective of the thesis to compare the developed trajectory model with real data, which omits the Coriolis effect. Coriolis force is therefore not taken into account for the purpose of this thesis.
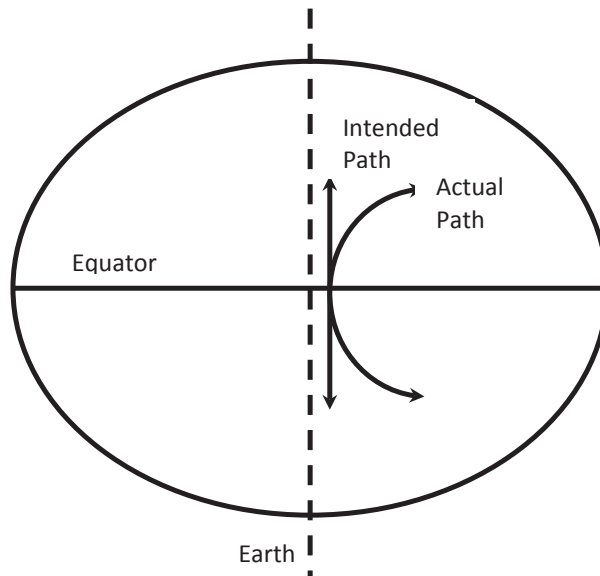
Figure 15. Coriolis Effect on Cross-Latitude Trajectory

5.   Magnus Effect

A spinning projectile at some relative velocity to the surrounding air experiences an additional force caused by unequal pressures on opposite sides of the spinning body. It is a result of viscous interaction between air and the spinning surface of the projectile. This effect is responsible for keeping a tennis ball in the air relatively longer when hit with an underspin or dropping faster when hit with a topspin. For instance, in Figure 16, the ball travels at some relative velocity to the air with a clockwise spin. Friction between the ball and air causes air to be dragged along in the same direction of the spin. As a result, the air just above it would have a lower relative velocity, and the air below it would slip by faster. The pressure above it would therefore be higher than the pressure below, resulting in a downward force.
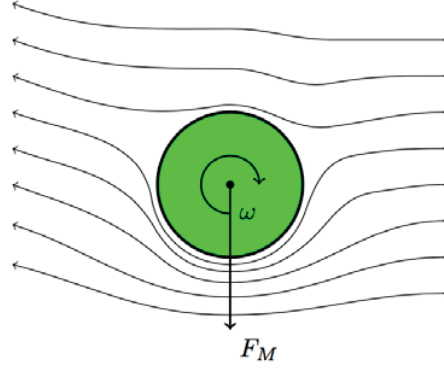
Figure 16. Flow Field around a Spinning Object where F_M Is the Resulting Magnus Force. Source: [11].

While this effect is significant in spherical objects, Skande [11] (Figure 17) has proven that the Magnus force for an artillery projectile is very small compared to drag and gravitational force and can be omitted. Hence, Magnus force is not considered for the purpose of this thesis.
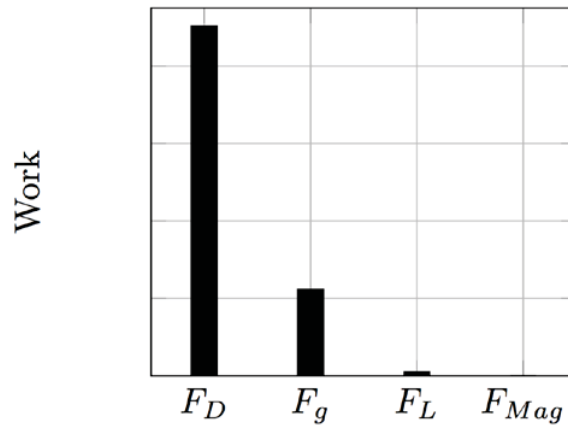
Figure 17. Total Work of Forces over Trajectory. Source: [11].

C.   TERMINAL BALLISTICS

Terminal Ballistics deals with events that take place when the projectile approaches the impact point. The ability of a projectile to damage a target depends on a variety of reasons, briefly,

1.	Impact Angle. The impact angle of the projectile affects its ability to penetrate a target. If the impact angle is small, the projectile may more likely ricochet, and any penetration will not be as deep as a projectile with steeper angle of impact since a projectile with a smaller angle of impact must travel through more material.

2.	Impact Velocity. The impact velocity affects the degree of penetration of the target. In general, a higher velocity projectile has more force and is thus able to penetrate deeper into targets. However, beyond a certain extent, the projectile may eventually shatter on impact.

3.	Shape and Weight. A slender and streamlined projectile experiences less drag and thus increased impact velocity and penetration. However, a blunt nose projectile has more strength at the tip and does not break as easily upon impact. A heavier projectile travels a shorter distance, considering all other factors constant, but provides better penetration depth. A balance of these factors is crucial in obtaining the desired terminal effect.

While it is desirable to have greater penetration depth when targeting material targets, the similarly configured projectile will have little effect against targets on the surface such as troops in the open. Thus, a variety of fuzes to produce a desired terminal effect have been developed. The following are several common types described briefly:

1.	Point Detonating. A point detonating fuze is configured to detonate immediately upon impact. This type of fuze is effective against light material and uncovered personnel.

2.	Proximity Action. A proximity fuze detonates when the distance to the target becomes smaller than some preset value. This is usually used to maximize fragmentation as the primary damage mechanism and is thus most effective against troops in the open or light materiel targets such as aircraft and missiles.

3.	Delay Action. A delay fuze only detonates after some penetration, thus has fewer fragmentation effects than proximity or point detonating fuzes. This type of fuze is usually effective against targets such as fortifications, dugouts, and buildings, where internal damage is more important that exterior damage.

Intuitively, the various fuze settings can have errors associated with them. For example, there is a finite uncertainty in the proximity sensor of a fuze, and therefore, it results in a small error in range and/or deflection from the desired detonation coordinates. These effects are not considered in the scope of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TRAJECTORY MODELS

A. POINT MASS, ZERO DRAG TRAJECTORY MODEL - 2 DOF, RANGE ONLY

The two-dimensional point mass, zero drag trajectory model is the simplest form of trajectory simulation where the projectile is represented by a single imaginary point in space with no length and experiences no air resistance throughout its flight. Only two dimensions are modeled in this case, the horizontal (x) and vertical (y) axes. The reference axes are located at the projectile centroid and are in the horizontal and vertical directions throughout the flight, as shown in Figure 18.
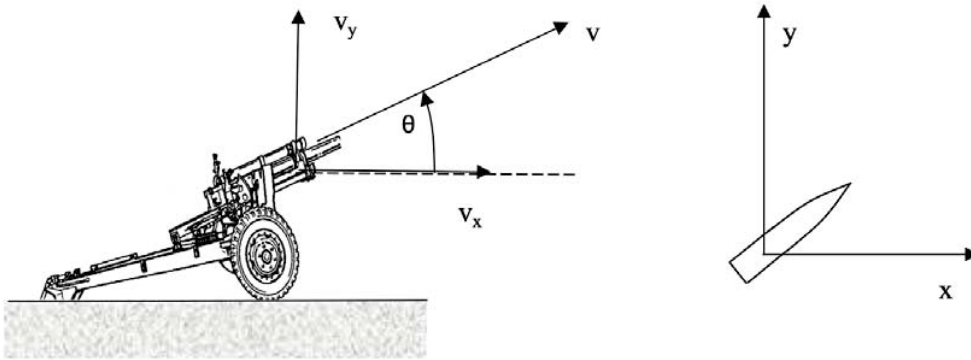
Figure 18. Reference Axes for 2D Point Mass, Zero Drag Trajectory Model.
Source: [4].

The initial energy input to the projectile produces the initial velocity and is only affected by gravity throughout the flight. The horizontal direction is not affected by gravity; therefore, its velocity component remains throughout the flight. The equations of motion for the vertical direction can be derived from Newton's Second Law for the following conclusions represented by Equations 13 and 14 [4]. This model results in closed form solutions that can be easily calculated.
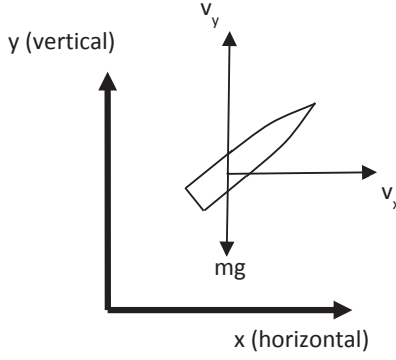
Figure 19. 2D Free Body Diagram for a Projectile with Zero Drag

$$v_x = v_{x,0}$$ (11)

$$x = v_{x,0}t$$ (12)

$$v_y = v_{y,0} + gt$$ (13)

$$y = v_{y,0}t + \frac{1}{2}gt^2$$ (14)

The time of flight can be obtained from solving the Equation 14 by setting y = 0:

$$TOF = \frac{-2v_{y,0}}{g}$$ (15)

The impact velocity is the root mean squared of the two velocity components:

$$v_{imp} = \sqrt{v_x^2 + v_y^2}$$ (16)

The impact angle is measured from the x-axis and can be evaluated from:

$$\theta_{imp} = \tan^{-1}\left(\frac{v_{y,imp}}{v_{x,imp}}\right)$$ (17)

where $v_{y,imp}$ is the y velocity component at impact, and $v_{x,imp}$ is the x velocity component at impact.

26

1. Model Inputs

The inputs to the trajectory model are shown in Table 1. At the elevation of 456 mils, the range according to the FT 155-AM-02 firing table is 15 km for a standard initial velocity of 684 m/s, produced from firing with charge 8.

Table 1.   Model Inputs for Point Mass, Zero Drag Trajectory Model

Parameters	Values
Firing elevation, θ_0	456 mils
Initial Velocity, v_0	684 m/s

It is customary for the artillery community to specify the firing angle, known as QE, in units of milliradians or mils. A complete 360-degree turn corresponds to 6400 mils. Hence, the conversion from mils to radians can be determined from the following equation:

$$\theta_{rad} = \frac{\pi \theta_{mils}}{3200}$$

(18)

2. Range for a Given Firing Angle

As shown in Figure 20, the resulting trajectory is a symmetric parabolic, where the apogee is precisely half of the range. The range obtained is 37,621 m, with an impact angle of 504 mils and impact velocity of 686.7 m/s. The time of flight is 60.4 sec.
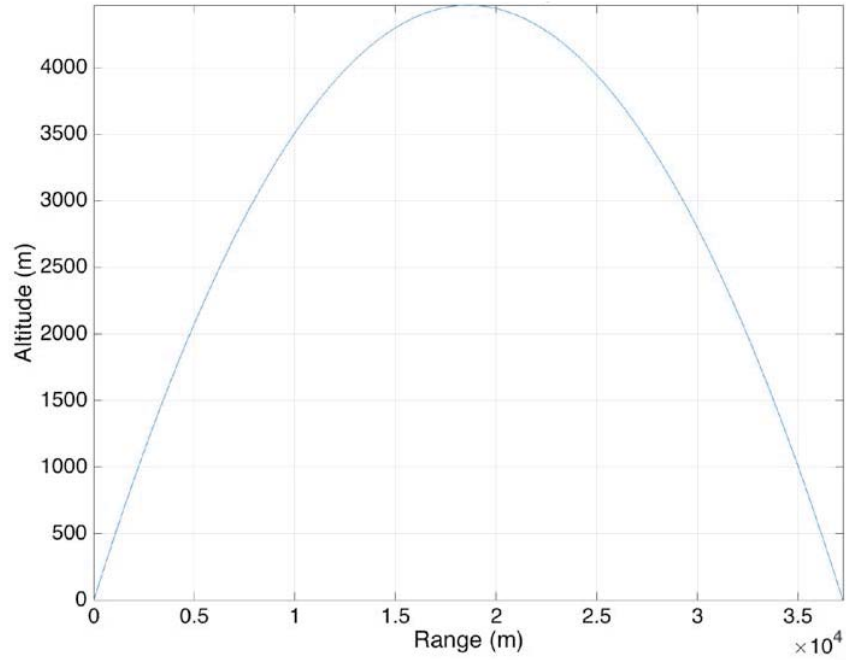
Figure 20. Trajectory Plot for Point Mass, Zero Drag Trajectory Model

3. Firing Angle for a Given Range

The required elevation to reach a desired range can be obtained from Equation 19. The solution produces two firing angles of θ and $\frac{\pi}{2} - \theta$, corresponding to low angle and high angle trajectories. These are shown in Figure 21.

$$\theta = \frac{1}{2}\sin^{-1}\frac{g.Range}{v_0^2}$$

(19)
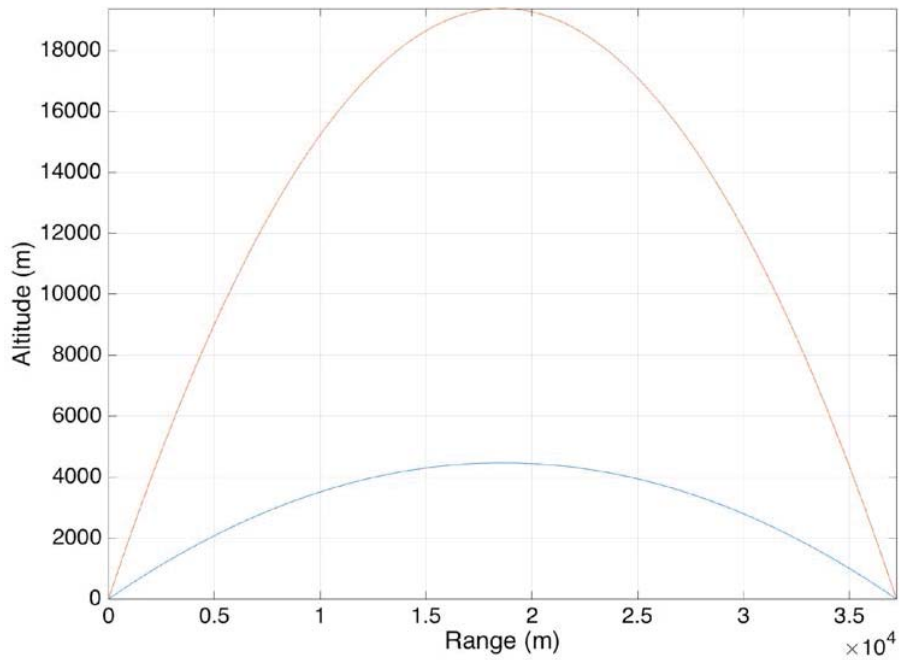
Figure 21. Low and High Angle Trajectories of Equal Range

4.   Firing Tables

For operational purposes, trajectory results are documented in books known as firing tables. These documents are specific to various combinations of charge and projectile types. Firing tables contain basic trajectory information and necessary corrections for non-ideal firing conditions, including wind, location, and drifting effects. Conventionally, these documents are used in the Battery Command Post to generate firing solutions manually in preparation for fire mission orders. This computation process is termed manual gunnery. In modern artillery, the data in firing tables is stored in the Fire Control Systems (FCS) where the firing solution can be generated much faster and minimizes human errors. However, the mathematical process of manual gunnery is still taught in military training today as a backup to the FCS. An example of the basic trajectory data extracted from a firing table is shown in Figure 22.

1	2	3	4	5	6	7	8	9
R A N G E	E L E V	FS FOR GRAZE BURST FUZE M564	DFS PER 10 M DEC HOB	DR PER 1 MIL D ELEV	F O R K	TIME OF FLIGHT	AZIMUTH CORRECTIONS	
							DRIFT (CORR TO L)	CW OF 1 KNOT
M	MIL			M	MIL	SEC	MIL	MIL
0	0.0			20	1	0.0	0.0	0.00
100	5.1			20	1	0.3	0.0	0.01
200	10.1			20	1	0.6	0.0	0.01
300	15.2			20	1	1.0	0.1	0.01
400	20.3			20	1	1.3	0.1	0.02
500	25.4			19	1	1.6	0.2	0.02
600	30.6	1.9	1.06	19	1	1.9	0.3	0.03
700	35.8	2.2	0.91	19	1	2.3	0.4	0.03
800	41.1	2.5	0.79	19	1	2.6	0.5	0.04
900	46.4	2.8	0.71	19	1	2.9	0.6	0.04
1000	51.7	3.2	0.63	19	1	3.2	0.7	0.04
1100	57.1	3.5	0.57	19	1	3.6	0.8	0.05
1200	62.5	3.8	0.53	18	1	3.9	0.8	0.05
1300	67.9	4.2	0.48	18	1	4.3	0.9	0.05
1400	73.4	4.5	0.45	18	1	4.6	1.0	0.06
1500	78.9	4.9	0.42	18	1	4.9	1.1	0.06

Figure 22. Basic Trajectory Data Extracted from a Firing Table. Source: [4].

The data from firing tables is derived from actual firings. From the example in Figure 22, column 1 shows the range of the projectile fired with an elevation listed in column 2. The corresponding total time of flight is shown in column 7. Azimuthal aiming corrections necessary to compensate for drifting effects are shown in column 8, and column 9 shows the azimuthal corrections needed to compensate for cross wind. A unique table exists for every charge increment of a particular combination of charge model and projectile.

5.   Discussion of Model Predictions

A comparison with the U.S. Artillery firing table FT 155-AM-02 for three sample ranges is shown in Table 2. It is observed that the impact velocity for the model is roughly equal to the initial velocity. The impact angle is lower due to the lower elevations needed to reach the same distance as compared to data from the

firing table. This also resulted in a lower max ordinate and shorter time of flight. For low drag munitions, the Point Mass, Zero Drag Trajectory Model gives results within 5 percent of higher fidelity models, but for high drag munitions, the errors increase considerably [4].

Table 2.   Comparison of Data from Point Mass, Zero Drag Trajectory Model with FT 122-AM-02

	QE (mils)	Impact Velocity (m/s)	Impact Angle (mils)	Max Ord (m)	TOF (s)
Fixed Range Comparison (5000m)					
Range Table	71	445	95	100	9.1
Zero Drag	53.49	684	53.5	65.7	7.32
Difference	-17.51	239	-41.5	-34.3	-1.78
% diff	**-24.66%**	**53.71%**	**-43.68%**	**-34.30%**	**-19.56%**
Fixed Range Comparison (8000m)					
Range Table	141.6	338	228	352	17
Zero Drag	85.84	684	85.87	168.95	11.74
Difference	-55.76	346	-142.13	-183.05	-5.26
% diff	**-39.38%**	**102.37%**	**-62.34%**	**-52.00%**	**-30.94%**
Fixed Range Comparison (10000m)					
Range Table	208.4	312	358	691	23.6
Zero Drag	107.58	684.01	107.71	265.01	14.7
Difference	-100.82	372.01	-250.29	-425.99	-8.9
% diff	**-48.38%**	**119.23%**	**-69.91%**	**-61.65%**	**-37.71%**

Clearly, the Point Mass, Zero Drag Trajectory Model is insufficient to predict real data for the artillery 155mm M107(HE). The primary reason is that drag force, a major parameter that affects the trajectory of the projectile, is not accounted for. In addition, as discussed in Chapter II, artillery projectiles are spin-stabilized, which produces an equilibrium yaw, causing a lateral force to be generated that results in the drifting of the projectile to the right.

B.   MODIFIED POINT MASS TRAJECTORY MODEL (MPMTM)

The Modified Point Mass Trajectory Model is a mathematical trajectory model used in the preparation of firing tables. The trajectory model is based on the equation of motion for a projectile and uses an estimate for the equilibrium yaw angle

to calculate lateral and vertical lift effects. As mentioned in Skande [11], the application of the MPMTM requires parameterization of a number of projectile data, such as aerodynamic coefficients and several form factors. These form factors are obtained from wind tunnel tests and by comparing trajectory model results and live firing tests. The equation of motion for the center of mass of the projectile for the MPMTM is shown in Equation 20 and illustrated by Figure 23. The evaluation of the forces in the equation of motion are not trivial because of coupling effects in range and deflection direction. This complete model is presented in Lieske [12].

$$m\vec{a} = \overrightarrow{F_d} + \overrightarrow{F_l} + \overrightarrow{F_g} + \overrightarrow{F_C} + \overrightarrow{F_M}$$

(20)

where F_d is the drag force, F_l is the lift force, F_g is the gravitational force, F_C is Coriolis force, and F_M is Magnus force.
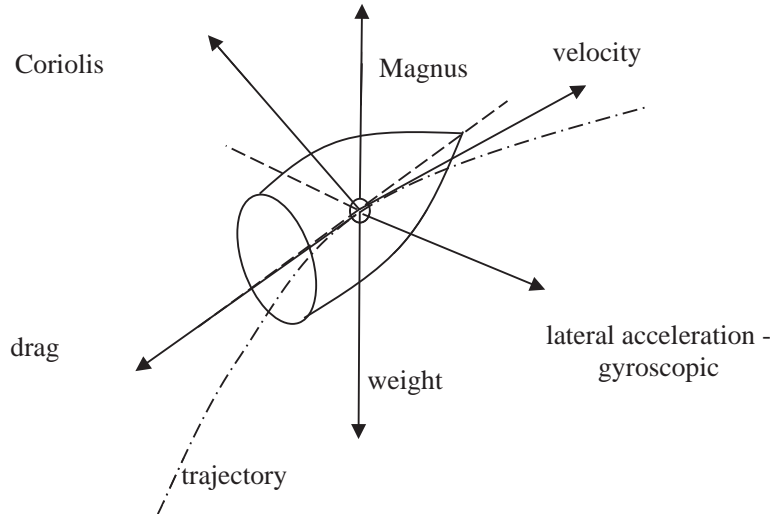
Figure 23. Forces and Motion of a Projectile. Source: [4].

C. INDIRECT FIRES DELIVERY ACCURACY PROGRAM (IFDAP) TRAJECTORY MODEL - 4 DOF, RANGE, AND DEFLECTION

The complete program developed in this thesis is called the Indirect Fires Delivery Accuracy Program (IFDAP). The trajectory model of the IFDAP is based on the MPMTM, omitting Coriolis and Magnus forces, and decoupling the motion in range from deflection. The side slip angle is considered for drift effects computation, and the angle of attack on range effects is considered sufficiently small to neglect. Second order drag and lift terms are also omitted to enhance computation efficiency.

These assumptions are validated by comparing trajectory results with the FT 155-AM-02.

1.   Reference Axes

The axes used is referenced to the projectile orientation at firing. This is the same as shown in Figure 18 with an additional z axis added in the lateral direction as shown in Figure 24. The angle θ is measured between the projectile's axis and the x-z plane and defines the pitch angle of the projectile. The side slip angle, β, is zero at launch, and the z-axis is pointed perpendicularly towards the right of the projectile (see Figure 24 and Figure 25).
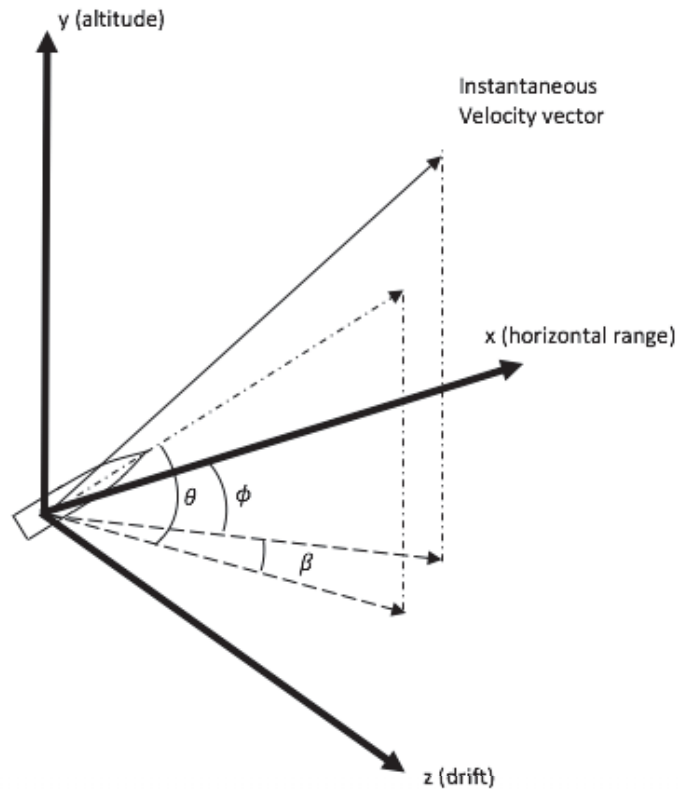
Figure 24. References Axes Definition in 3D
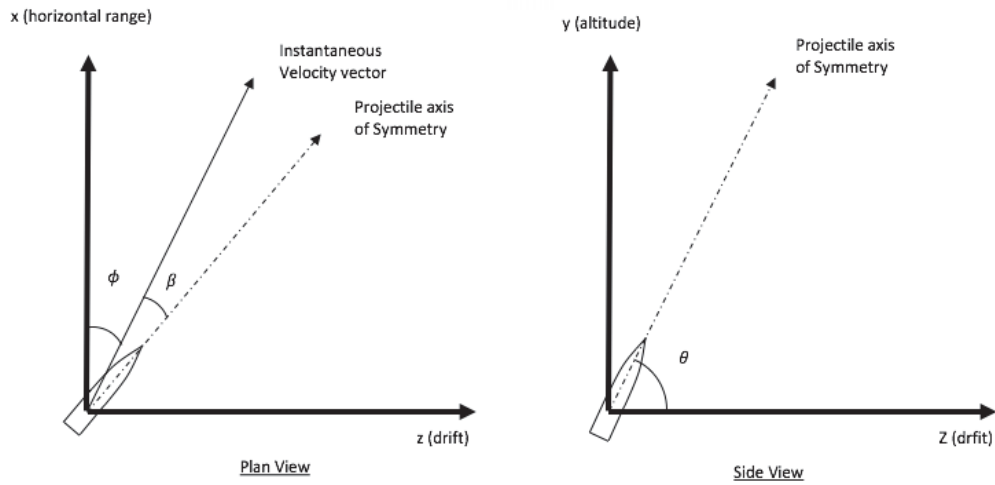
Figure 25. References Axes Definition in 2D

2. Acceleration

Since each axis of motion (horizontal, vertical, and lateral) is treated independently, the drag force needs to be resolved for each independent axis to compute the corresponding acceleration (Figure 26). These are shown in Equations 21 to 24.
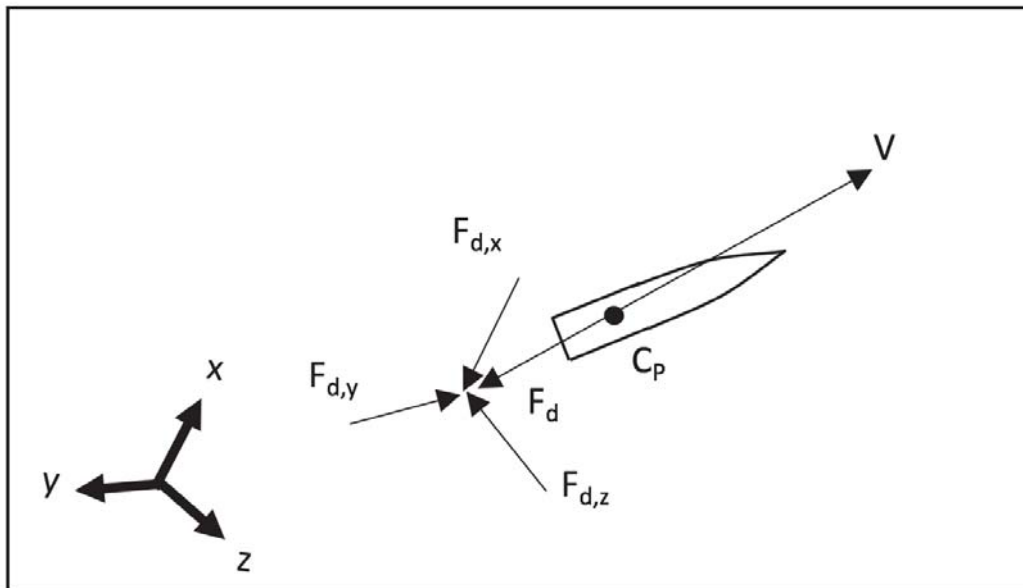
Figure 26. Orthogonal Components of Drag Force

$$F_{d,x} = F_d \cos\theta \cos\phi \tag{21}$$

where θ is obtained from Equation 22,

$$\phi = \tan^{-1}\left(\frac{v_z}{v_x}\right) \tag{22}$$

$$F_{d,y} = F_d \sin\theta \tag{23}$$

$$F_{d,z} = F_d \cos\theta \sin\phi \tag{24}$$

The acceleration in each axis is derived from equating Newton's Second Law with the drag force equation. For the horizontal (x) and vertical (y) axes, the results are as follows:

$$a_x = -\frac{\frac{1}{2}SC_d\rho V^2 \cos\theta \cos\phi}{m} \tag{25}$$

$$a_y = g - \frac{\frac{1}{2}SC_d\rho V^2 \sin\theta}{m} \tag{26}$$

In the lateral (z) direction, the resulting force is a sum of the lift force (Equation 10) that acts in the positive z-direction, and a drag force that acts in the negative z-direction.

$$a_z = \frac{F_{l,z} - F_{d,z}}{m} \tag{27}$$

If the normal force cofficient is used in evaluting the lift force, the resulting accerlation is normal to the projectile axis of symmetry; hence, the z-component can be evaluted from multiplying the lift force by $\cos(\phi+\beta)$. The resulting acceleration in the z-direction is:

$$a_z = \frac{1}{2}S\rho V^2 \frac{(C_{N_\alpha}\sin\beta\cos(\phi+\beta) - C_d\cos\theta\sin\phi)}{m} \tag{28}$$

where ϕ is evaluated from equation 22, and β is evaluated from Equation 6.

35

3. Velocity

The acceleration computed from Equations 25 to 27 is taken as the forward difference of the velocity divided by the simulation time step. Taking the horizontal (x) component as an example, the velocity can be evaluated from Equations 29 and 30. This is similarly done for the vertical (y) and lateral (z) axes, and angular velocity (p).

$$a_x = \frac{v_x(t+dt)-v_x(t)}{dt}$$
(29)

$$v_x(t+dt)=v_x(t)+a_x(t)dt$$
(30)

$$v_y(t+dt)=v_y(t)+a_y(t)dt$$
(31)

$$v_z(t+dt)=v_z(t)+a_z(t)dt$$
(32)

This results in v_x, v_y, v_z, and p at t + dt. The total velocity of the projectile can be determined from the root-sum-squared of all othorgonal velocity components:

$$V = \sqrt{v_x^2+v_y^2+v_z^2}$$
(33)

4. Displacement

Following velocity calculations, the displacement can be computed using the same method:

$$x(t+dt)=x(t)+v_x(t+dt)dt$$
(34)

$$y(t+dt)=y(t)+v_y(t+dt)dt$$
(35)

$$z(t+dt)=x(t)+v_z(t+dt)dt$$
(36)

This results in x, y, z, at t + dt.

5.   Angular Calculations

Using the orthogonal velocity components, the angles that are used to resolve the forces and calculate the lateral lift can be calculated.

a.   *Projectile Pitch Angle*

Extending Equation 17 to three dimensions, the pitch angle of the projectile at t + dt can be calculated from:

$$\theta(t+dt) = \tan^{-1}\left(\frac{v_y(t+dt)}{\sqrt{v_x^2(t+dt)+v_z^2(t+dt)}}\right)$$

(37)

b.   *Side Slip Angle*

From Equation 6, the side slip angle at t + dt can be evaluated using:

$$\beta(t+dt) = \frac{8p(t+dt)I_{xx}g\cos\theta(t+dt)}{\pi p D^2 L_{ref}C_{m_a}V(t+dt)^3}$$

(38)

c.   *Angle between Velocity Vector and x-axis*

From Equation 22, the angle between the velocity vector and the x-axis at t + dt can be evaluated is:

$$\varphi(t+dt) = \tan^{-1}\left(\frac{v_z(t+dt)}{v_x(t+dt)}\right)$$

(39)

The solution to the next time step may be accomplished by setting variables at (t + dt) to t and repeating calculations from sections 1 through 5.

6.   Aerodynamic Coefficients

Since the altitude of the projectile changes at every time step, the speed of sound is re-evaluated to determine the correct areodynamic coefficients C_{m_a}, C_{N_a}, C_{l_p}, C_d, and air properties, T and ρ. Equation 40 is used to determine the speed of

sound. The equations used to interpolate the aerodynamic coefficients and air properties are presented in the subsequent sections.

$$v_M = \sqrt{kRT}$$

(40)

a. Interpolation of Air Properties

U.S. standard atmosphere air properties are used in the trajectory computation. From the raw data of properties, two constants, a_0 and a_1, are calculated. A straight line is assumed between two consecutive discrete set of air properties. The straight line would have a Y-intercept, a_0, and gradient, a_1, determined from the raw data. The interpolation is done using the following equation:

$$Y = a_0 + a_1 X$$

(41)

In the interpolation of air properties, the dependent variables are air temperature and air density, and the independent variable is the altitude of the projectile. At the current altitude of the projectile, the independent variable, X (altitude), falls within X_1 and X_2, the two altitude lines below and above the current altitude in the raw data. The a_0 and a_1 corresponding to X_2, the altitude line above, are used to interpolate the dependent variable. An example is shown in Table 3.

Table 3.   Example of Altitude vs. Air Temperature with
Various a_0 and a_1 Values

Geo potential Altitude above Sea Level	Temperature	a0	a1
(m)	(K)		
2000	275.15	288.15	-0.0065
3000	268.66	288.13	-0.00649

In this example, the dependent variable is temperature and the independent variable is the altitude of the projectile. Suppose the current altitude of the projectile is at 2500 m. This altitude falls between $X_1 = 2000$ m and $X_2 = 3000$ m. The a_0 and a_1 values are therefore 288.13 and -0.00649. Hence, the air temperature at 2500 m is:

38

$$AirTemp = 288.13 + (-0.00649)(2500) = 271.91K \qquad (42)$$

b.   Interpolation of Aerodynamic Properties

The interpolation of aerodynamic properties uses the same method presented for the interpolation of air properties. The instantaneous Mach number of the projectile is evaluated by dividing Equation 33 by Equation 40.

$$M = \frac{V}{v_m}$$

(43)

7.   Termination of Simulation

From the aforementioned process, at each time increment, we are calculating the accerlation variables a_x, a_y, a_z, and $\dot{p}$; the velocity variables v_x, v_y, v_z; and p; the displacement variables x, y, z; and the angular variables θ, φ, and β. The simulation ends where the altitude, y, becomes negative, and the impact conditions are the last values determined. The list of output variables is shown in Table 4.

Table 4.   Table of Output Variables

Terminal Output Variables	Values
Quadrant Elevation	θ_0
Range	$\sqrt{x(t+dt)^2 + y(t+dt)^2 + z(t+dt)^2}$
Initial Velocity	V_0
Impact Velocity	$\sqrt{v_x(t+dt)^2 + v_y(t+dt)^2 + v_z(t+dt)^2}$
Drift (m)	z(t+dt)
Drift (mils)	$\approx \dfrac{z(t+dt)}{range/1000}$
Max Ordinate	Maximum of y
TOF	t+dt

8. Wind Effects

The modeling of wind effects is limited to range wind (x-axis) and cross wind (z-axis) as vertical wind usually has insignificant effects. The effects of wind are accounted for by modifying the projectile velocity components to relative velocity to the air. This affects the projectile Mach number and the areodynamic coefficients used. Specialized artillery units are often dedicated to obtaining metrological data in the vicinity of firing by sending a sounding balloon, which carries instruments on board to transmit a profile of weather parameters including air humidity, temperature, wind speed, atmospheric pressure, by means of a measuring device called a *radiosonde*.

a. *Range Wind Effects*

The effect of range wind can be understood by accounting for the relative change in velocity of the projectile relative to the air (Figure 27). In the case of no range wind, the velocity of the projectile relative to air in the horizontal (x) direction, $v_x^{P/A}$, is just the velocity of the projectile relative to the ground, $v_x^{P/G}$ or v_x, which is positive in the x-direction. With the effects of range wind, this vector is redefined using the kinematic relative velocity equation [4]:

$$v_x^{P/A} = v_x^{P/G} + v_x^{G/A} = v_x - v_x^{A/G}$$

(44)

where $v_x^{A/G}$ is positive for tail wind and negative for head wind.
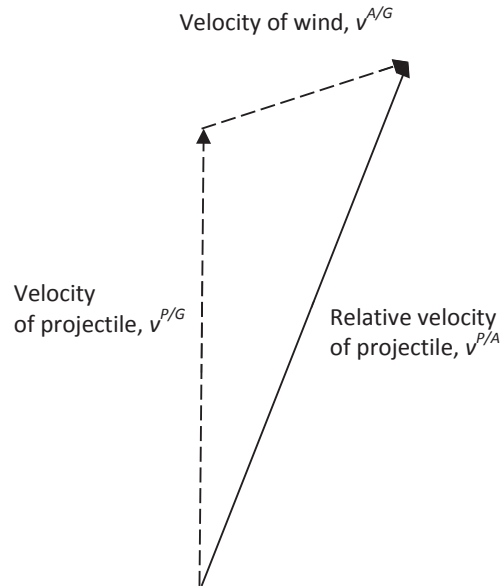
Figure 27. Relative Velocity of Projectile

b.   *Cross Wind Effects*

The effect of cross wind is accounted for using the same methodology as described previously for range wind.

9.   Initial Comparison between Models

The inputs to the IFDAP trajectory model are shown in Table 5. At the quadrant elevation of 456 mils, the range according to the FT 155-AM-02 firing table is 15 km for a standard initial velocity of 684 m/s, when firing with charge 8.

Table 5.   Model Inputs for IFDAP

Parameters	Values
Firing elevation, θ_0	456 mils
Initial Velocity, v_0	684 m/s

A trajectory comparison between the FT 155-AM-02, IFDAP, and Zero Drag Model is shown in Table 6. Contrary to the Point Mass, Zero Drag Trajectory Model, the trajectory obtained from the IFDAP is not parabolic. The apogee is skewed towards the second half of the projectile range, and the range is significantly shorter (Figure 28 and Figure 29). The angle of fall is also steeper than the zero drag model.

41

The shape of the trajectory is due to the fact that most of the energy imparted at ignition is lost during the ascending phase of the flight. Once past the apogee, the pitch angle of the projectile decreases quickly, leading to a higher impact angle than the zero drag model. This result indicates the significance of the drag force.

Table 6.   Trajectory Comparison between FT 155-AM-02, IFDAP and Zero Drag Model

Variables	FT 155-AM-02	IFDAP	Zero Drag Model
Range	15,000 m	14,884 m	37, 621 m
Impact Angle	709 mils	707 mils	504 mils
Impact Velocity	314 m/s	314.5 m/s	686.7m/s
Time of Flight	43.7 s	43.27 s	60.4 s
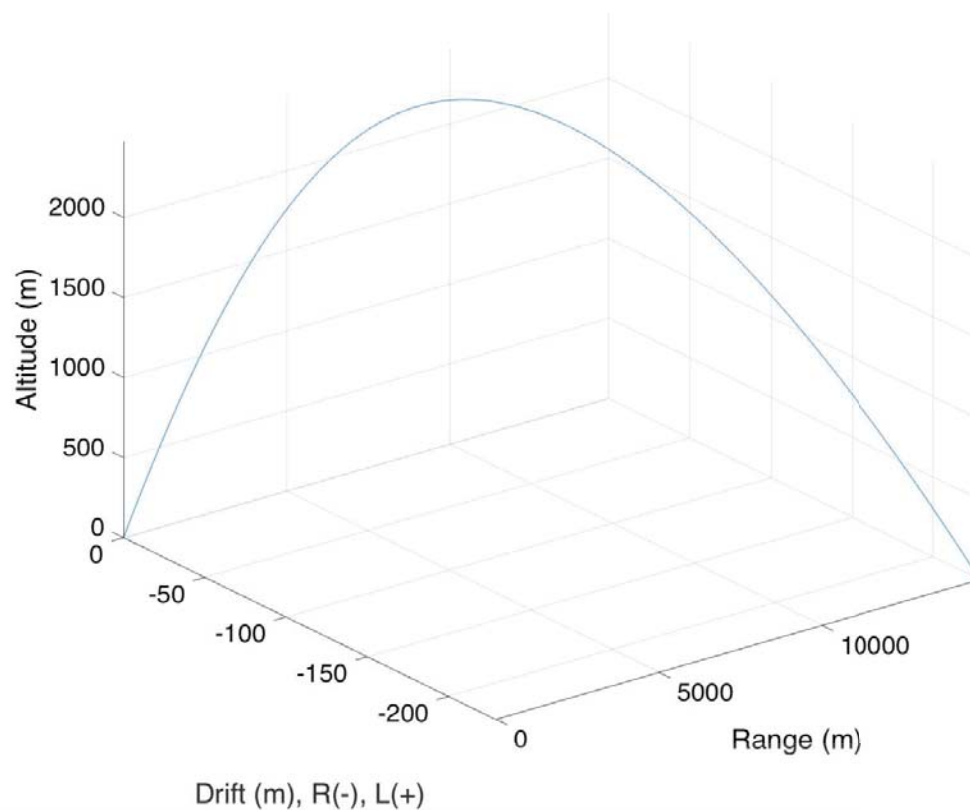
Figure 28. IFDAP Trajectory Model
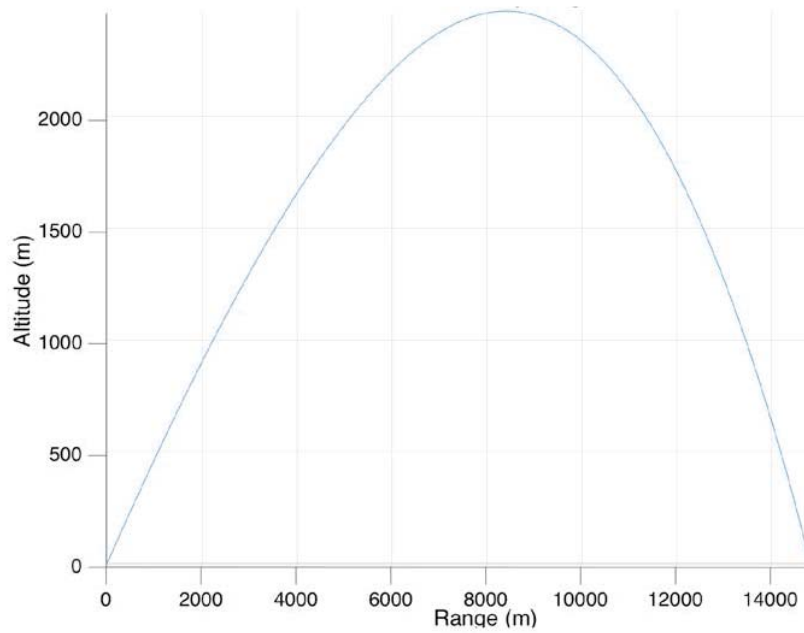
Figure 29. 2D Trajectory Plot (Range/Altitude) from IFDAP Model

10. Discussion

A more detailed comparison with the U.S. Artillery firing tables is shown in Table 7. The IFDAP based on the MPMTM clearly provides a much better estimation of the outputs than the simple zero drag trajectory model. These simplified equations of motion enable rapid computation, and their degree of accuracy is sufficient for conventional artillery area target missions. Comparing the FT 155-AM-02 with the IFDAP trajectory model, the range estimation is within 1 percent; hence, this proves the increase in range due to the angle of attack component in the yaw of repose is sufficiently small to be neglected for computational efficiency.

Table 7.   Comparison of Data between IFDAP Trajectory Model and
FT 155-AM-02

	Range (m)	Initial Velocity (m/s)	Impact Velocity (m/s)	Impact Angle (mils)	Drift (m)	Drift (mils)	Max Ord (m)	TOF (s)	Max Range (m)
Fixed QE Comparison (71 mils)									
Range Table	**5000**	684	445	95	11	2.2	100	9.1	18100
Thesis	4954.13	447.14	94.11	11.36	2.29	99.12	8.99	17752	17752
Difference	-45.87	2.14	-0.89	0.36	0.09	-0.88	-0.11	-348	-348
% diff	-0.92%	0.48%	-0.94%	3.27%	4.09%	-0.88%	-1.21%	-1.92%	-1.92%
Fixed QE Comparison (208.4 mils)									
Range Table	**10000**	684	312	358	71	7.1	691	23.6	
Thesis	9921.19	312.91	355.87	69.28	6.98	683.75	23.34	9921.19	
Difference	-78.81	0.91	-2.13	-1.72	-0.12	-7.25	-0.26	-78.81	
% diff	-0.79%	0.29%	-0.59%	-2.42%	-1.69%	-1.05%	-1.10%	-0.79%	
Fixed QE Comparison (455.9 mils)									
Range Table	**15000**	684	314	709	234	15.6	2497	43.7	
Thesis	14882.53	314.44	706.78	228.11	15.33	2475.62	43.26	14882.53	
Difference	-117.47	0.44	-2.22	-5.89	-0.27	-21.38	-0.44	-117.47	
% diff	-0.78%	0.14%	-0.31%	-2.52%	-1.73%	-0.86%	-1.01%	-0.78%	

IV. ERROR AND ACCURACY COMPUTATIONS

A. ERROR BUDGETS AND BALLISTIC PARTIALS

The sets of parameters used in the error models to estimate precision and MPI errors are the ballistic partials and error budgets. The ballistic partials are also known as unit effects; these are the change in the dependent variable for a unit change in the independent variable, and the error budgets are the standard deviations in the independent variable. The two quantities multiply to determine the standard deviation in the dependent variable, as shown in equation 46.

One of the factors that affect the range of a projectile is the muzzle velocity of the gun. Suppose all other factors are kept constant, we can express the unit change in range due to a change in muzzle velocity as [4]:

$$\frac{\Delta x}{\Delta V} = \frac{\delta x}{\delta V} \tag{45}$$

Thus,

$$\Delta x = \frac{\delta x}{\delta V} \Delta V \tag{46}$$

Computing the unit effect $\frac{\delta x}{\delta V}$ is be done by running the trajectory model with a small perturbation, about the nominal firing solution dv, and observing the corresponding change in impact point, dx. In general, the smaller the perturbation, the more accurate the unit effect computed. For the purpose of this thesis, the same perturbations used in the production of the firing table FT 155-AM-02 is used for the computations of the partials. These are summarized in Table 8.

Suppose multiple firings over multiple occasions are done and records of the variations in muzzle velocity are made. Using a chi-squared test, we will observe that the muzzle velocity is normally distributed [4]. We can compute a standard deviation, σ_V, and subsequently compute a standard deviation in range, σ_x, using the following equation. The quantity σ_V is known as the error budget.

$$\sigma_x = \frac{\delta x}{\delta V} \sigma_V \tag{47}$$

The combination of multiple factors on the range is the root sum squared of the factors.

Table 8.   Perturbations Used for Computing Ballistic Partials

Factors	Perturbation
QE	50 mils
Muzzle Velocity	10 m/s
Ballistic Air Temperature	10%
Ballistic Air Density	10%
Range and Cross Winds	50 Knots

B.   ERROR MODELS

1.   Precision Error

The precision error is a measure of the dispersion of a group of rounds fired by a single gun about the mean point of impact on a single occasion. This model assumes that the muzzle velocity, drag and quadrant elevation are the three most significant factors that affect the range dispersion. The precision errors in range and deflection can be estimated using equations 48 and 49 respectively [4].

$$\sigma_{Px}^2 = \left(\frac{\delta x}{\delta V} \sigma_V \right)^2 + \left(\frac{\delta x}{\delta Drag} \sigma_{Drag} \right)^2 + \left(\frac{\delta x}{\delta \theta} \sigma_\theta \right)^2$$

(48)

$$\sigma_{Pz}^2 = \left(\frac{a_0}{1018.59} \times \frac{a_1 x}{a_1 - \theta} \right)^2$$

(49)

where $\dfrac{\delta x}{\delta V}$ Partial of range with muzzle velocity

$\dfrac{\delta x}{\delta Drag}$ Partial of range with air drag coefficient

$\dfrac{\delta x}{\delta \theta}$ Partial of range with QE

a. *Error Budgets in Precision Error Model*

The error budget in muzzle velocity, σ_V, is dependent on the internal ballistics that affect the muzzle velocity. This includes factors related to the barrel, projectile and the charge.

The error budget in ballistic coefficient, σ_{Drag}, is a measure of how well we know the Mach number vs C_d relationship.

The error budget in QE, σ_θ, is a measure of the variation between rounds in the true QE when the projectile leaves the barrel. This error can be caused by undesired barrel jumps, as explained in chapter 2, slight displacement of the gun platform due to recoil from continuous firing, or a mechanical error causing the elevation to slip slightly from continuous firing.

In the computation of the partial in range with ballistic air drag coefficient, $\dfrac{\delta x}{\delta Drag}$, the variation in air density is used instead of the drag coefficient C_d. Since air density and directly proportionate to C_d, their partials are the same.

We assume the following error budgets in Table 9 as representative values.

Table 9. Values of Error Budgets and Parameters Used for Precision Error Computations

Equation Parameters	Values
σ_v	0.92 m/s
σ_{Drag}	0.65%
σ_θ	0.3 mils
a_0	0.52
a_1	2000

2. **Mean-Point-of-Impact (MPI) Error**

The Mean-Point-Of-Impact (MPI) error is a measure of the variability of the mean point of impact of a single gun, aiming at the same target coordinates, over multiple occasions. This means that the gun had to be re-aimed at the target between occasions, and therefore error budgets to account for this re-aiming error needs to be

considered. The error budgets in QE and velocity are used to quantify this error. In addition, the target and gun location will also have some uncertainty associated with it. Since the amount of error in location accuracy has the same effect in the overall MPI range and deflection errors, the partials for location accuracies are unity.

$$\frac{x}{x_{Loc}} = \frac{z}{z_{Loc}} = 1$$

(50)

Unlike precision error, the MPI error is dependent on the technique used. There exist different sets of equations applicable for the different techniques of firing. In this thesis, the error models will be based on the predict fire technique. The following equations presented are for the case of predicted fire using point detonating fuze.

$$\sigma_{MPI,x}^2 = \left[\frac{\delta x}{\delta \rho}\right]^2 (\sigma_\rho^2 + \sigma_{drag}^2) + \left(\frac{\delta x}{\delta T}\sigma_T\right)^2 + \left(\frac{\delta x}{\delta w}\sigma_w\right)^2 +$$
$$\left(\frac{\delta x}{\delta V}\sigma_V\right)^2 + \left(\frac{\delta x}{\delta \theta}\sigma_{AIM-EL}\right)^2 + \sigma_{LOC-x}^2 + \sigma_{CHART-x}^2$$

(51)

$$\sigma_{MPI,z}^2 = \left(\frac{\delta z}{\delta w}\sigma_w\right)^2 + \left(\frac{\delta z}{\delta LIFT}\sigma_{LIFT}\right)^2 +$$
$$\left(\frac{\delta z}{\delta a}\sigma_{AIM-AZ}\right)^2 + \sigma_{LOC-z}^2 + \sigma_{CHART-z}^2$$

(52)

where $\dfrac{\delta x}{\delta \rho}$ Partial in range with air density

$\dfrac{\delta x}{\delta T}$ Partial in range with air temperature

$\dfrac{\delta x}{\delta w}$ Partial in range with wind velocity

$\dfrac{\delta z}{\delta w}$ Partial in deflection with wind velocity

$\dfrac{\delta z}{\delta LIFT}$ Partial in deflection with lift force, determined from [4]:

$$\frac{\delta z}{\delta LIFT} = \frac{Drift}{100} \times \frac{range}{1018.59}$$

$$(53)$$

where Drift is the drift in mils calculated from drift in meters as follows:

$$Drift(mils) = z \times \frac{range}{1018.59}$$

$$(54)$$

$\dfrac{\delta z}{\delta a}$ Partial in deflection with azimuth angle, determined from [4]:

$$\frac{\delta z}{\delta a} = \frac{range}{1018.59}$$

$$(55)$$

a.   *Error Budgets in MPI Unadjusted Fire Model*

The lift error budget, σ_{LIFT}, is a measure of how well we know the relation of lift coefficient C_{L_a} with Mach number.

The meteorological error budgets, σ_T, σ_w and σ_ρ are not fixed and depends on the duration from which the last met message was updated and the distance of the gun from the position where the met data was collected. These variables are referred to as the MET errors and they depend on the time and distance between measuring MET conditions and firing. The longer the staleness hour and/or separation distance, the more likely the meteorological data will be erroneous, hence higher error budgets. The variation in met error budgets with staleness hour and distance is shown in Appendix C.

The error budget in aiming error for QE and azimuth, σ_{AIM-EL} and σ_{AIM-AZ}, accounts for the crew competency in aiming the gun, mechanical and instrumentation error of the gun between occasions.

The error budget, $\sigma_{CHART-x}$ and $\sigma_{CHART-z}$, is the measure of uncertainty in the target location. Target location may be obtained by map spotting, forward observers on the ground, or from UAVs with targeting sensors. Different techniques will have different accuracies associated with them. The most accurate method is to use a GPS on the target location to determine the coordinates. However, this is not practicable and the next best method would be to have a forward observer nearby on the ground using an accurate laser range finder.

The error budget in location accuracy for range and deflection, σ_{LOC-x} and σ_{LOC-z}, is a measure of the uncertainty in the gun location. Conventional methods use marked survey control points to ensure all pieces lie on common grid (i.e. common error). Modern Artillery pieces are positioned using GPS/INS systems that effectively reduce this error to a small amount.

Table 10 shows all the values of error budgets used for the MPI unadjusted fire error model.

Table 10. Values of Error Budgets and Parameters Used for MPI Error Computations

Equation Parameters	Values
σ_{LIFT}	1
σ_v	3 m/s
σ_{Drag}	1%
σ_θ	0.5 mils
$\sigma_{azimuth}$	1 mils
σ_{wind}	As accordingly
σ_p	As accordingly
σ_T	As accordingly
$\sigma_{chart-X}$	0 m
$\sigma_{chart-Z}$	0 m
σ_{Loc-X}	15 m
σ_{Loc-Z}	15 m

C.   MEASURES OF DISPERSION

It is possible to calculate a mean and variance for range and deflection if we measure all range and deflection deviations of each round. However, this is only valid for a single gun. In a typical Artillery mission, a battery of 6 guns is used, firing up to 10 rounds each. In order to more accurately define the errors, two quantities were introduced in chapter I; a dependent error, and an independent error, or MPI and Precision Error, otherwise known as aiming error and ballistic dispersion within the air community. The MPI error is the ability of the crew to aim the gun correctly at the target, whereas the precision error is attributed to round-to-round dispersion. We can define the range (x) and deflection (z) total miss distances from the desired impact point as follows:

50

$$x_{miss,TOT} = x_{miss,MPI} + x_{miss,Prec}$$ (56)

$$z_{miss,TOT} = z_{miss,MPI} + z_{miss,Prec}$$ (57)

Since the precision and MPI errors are independent random variables, the root-sum-square of their standard deviation will give the standard deviation of the total miss distance [4].

$$\sigma^2_{TOT,x} = \sigma^2_{prec,x} + \sigma^2_{MPI,x}$$ (58)

$$\sigma^2_{TOT,z} = \sigma^2_{prec,z} + \sigma^2_{MPI,z}$$ (59)

D.   PROBABLE ERROR

The Probable Error (PE) is a statistical quantity which is defined as the interval from the desired point of impact (DPI) which would encompass 50% of all projectiles fired aiming towards it. There are three ways of defining this metric in this thesis: 1) Range Error Probable (REP), 2) Deflection Error Probable (DEP), and 3) Circular Error Probable (CEP).

The range and deflection probable error of a normally distributed random variable is related to the standard deviation by the following equation:

$$PE = 0.6745\sigma$$ (60)

For certain applications, it may be more convenient to specify a single error probable value determined by the absolute miss distance from the DPI, rather than in terms of range and deflection. This value is the circular error probable, and can be computed using the standard deviations in range and deflection derived from equations 58 and 59 as follows:

$$CEP = 1.1774\frac{(\sigma_x + \sigma_y)}{2}$$ (61)

Note that CEP is not a good measure of dispersion. Two different weapons can have the same CEP by one can be high in REP but low in DEP and the other can be high in DEP but low in REP. They would result in very different fractional damage for different dimensions of area of target elements. Hence it is more useful to use the

accuracy metrics $\sigma^2_{TOT,x}$ and $\sigma^2_{TOT,z}$, which characterizes the dispersion of the gun. The CEP metric is only meaningful if the distribution of the impact points is circularly normal.

V. IMPLEMENTATION AND CASE STUDY

The trajectory and accuracy models described in Chapters III and IV were initially coded in MATLAB because of its computational capability and simple graphing techniques. In this chapter, a case study is demonstrated with the objective of answering the following question: "What is the accuracy (MPI and Precision) of an M107 fired at a range of 15 km with a muzzle velocity of 684m/s ?"

A. MODEL SELECTION AND INPUTS

The IFDAP trajectory model described in Chapter III is selected for the implementation. The aerodynamic coefficients used are based on Table 11. This table represents a typical dataset of aerodynamic coefficients available for a projectile. Between each discrete value of Mach number, we interpolate the aerodynamic coefficients using the method described in Chapter III Section C Subsection 6.

Table 11. Aerodynamic Coefficients of M107 (HE) Projectile. Source: [13].

M	C_A	$C_{A_{\alpha^2}}$	C_{N_α}	$C_{Y_{p\alpha}}$	C_{l_p}	C_{m_α}	C_{m_q}	$C_{n_{p\alpha}}$			
								0°	2°	5°	10°
0.01	.144	2.343	-1.763	-0.767	-.023	3.355	-5.1	-0.500	0.005	0.294	0.58
0.60	.144	2.343	-1.763	-0.767	-.023	3.378	-5.1	-0.500	0.005	0.294	0.58
0.80	.146	2.847	-1.783	-0.767	-.022	3.571	-5.1	-0.355	0.078	0.366	0.65
0.90	.167	3.372	-1.827	-0.857	-.021	3.957	-7.4	-0.112	0.172	0.415	0.86
0.95	.221	3.73	-2.038	-1.082	-.020	3.886	-9.9	0.085	0.292	0.500	1.12
1.00	.327	4.180	-2.153	-0.992	-.020	3.682	-13.8	0.198	0.388	0.482	0.72
1.05	.383	4.691	-2.207	-0.902	-.020	3.415	-13.3	0.293	0.430	0.465	0.55
1.10	.381	5.209	-2.255	-0.857	-.019	3.384	-14.6	0.334	0.432	0.456	0.54
1.20	.370	5.702	-2.325	-0.767	-.020	3.424	-15.8	0.352	0.424	0.438	0.51
1.35	.353	5.130	-2.442	-0.767	-.020	3.278	-15.6	0.366	0.424	0.438	0.51
1.50	.338	4.561	-2.556	-0.767	-.020	3.264	-15.3	0.373	0.424	0.438	0.51
1.75	.314	3.970	-2.692	-0.767	-.020	3.201	-15.3	0.381	0.431	0.438	0.51
2.00	.294	3.460	-2.747	-0.767	-.021	3.013	-15.3	0.388	0.431	0.438	0.51

The weapon data and general parameters required are shown in Table 12. The launch altitude of the weapon and target height are assumed to be at the same level, and there is no wind simulated. A MET staleness of 0.5 hour is assumed.

Table 12. General Parameters and Weapon Data Used in Case Study

General Parameters		Weapon Data	
Time Step	0.01 s	Twist Rate	20 Calibers/rev
g	9.81 m/s^2	I_{xx}	0.1461 Kg.m^2
Air C_p	1005 J/Kg.K	Mass	43.091 Kg
Air C_v	718 J/Kg.K	Projectile Diameter	0.155 m
F_l	0.9076		
F_T	1		
F_ρ	1		

B.   QE FINDER

Given the QE, it is easy to find the range at impact using the IFDAP trajectory model. The opposite, however, is not trivial. In reality, target acquisition elements such as the FO obtain the range and direction to the target using a laser range finder and transmit this data to artillery elements. The fire control system has a database of the firing table and searches for the required QE to reach the required range. In the IFDAP, the required QE to reach the desired range is narrowed down through an iterative process. The user is first required to input four parameters: 1) a desired range, 2) muzzle velocity, 3) the required accuracy, i.e., how close to the desired range is acceptable to the user, and 4) the number of iterations. Using the desired range and muzzle velocity specified by the user, an initial QE is computed using the zero drag model from Equation (19). This zero drag QE is first used to compute the range using the IFDAP trajectory model. Using the QE found from the zero drag model on the IFDAP trajectory model will certainly result in the computed range falling short of the desired range due to drag; hence, a new QE(i+1) is generated using the following equation:

$$QE(i+1) = QE(i) + \frac{\Delta Range}{dR/dQE}$$

(62)

where ΔRange is the difference in desired range and computed range, and

dR/dQE is the partial in range with QE.

The new QE is input to the trajectory computer to calculate the new range. This loop continues as long as both of the following conditions are not met:

Did the range meet the required accuracy?
OR
Is the current number of iterations (i) more than specified?

Once either of the conditions is met, the QE Finder procedure ends, and the calculated range and QE are written in the output cells. A flow diagram of the QE Finder model is shown in Figure 30.
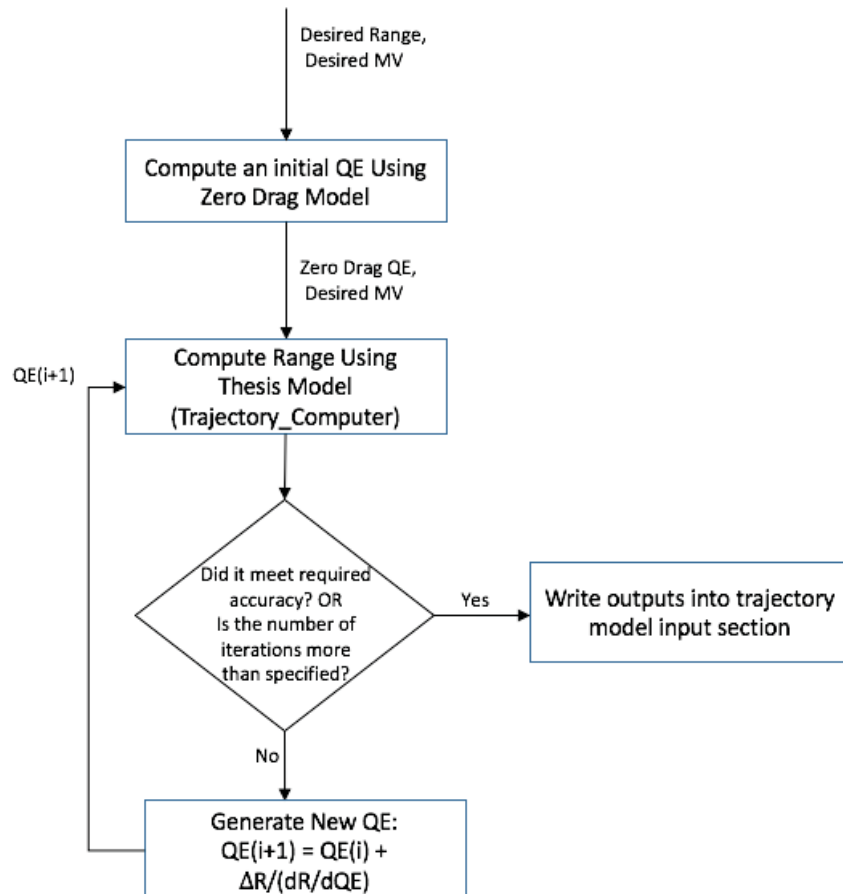
Figure 30. Computation Flow Diagram of QE Finder Model

C.   ACCURACY MODEL

The overall model to compute the accuracy metrics is presented as a flow diagram in Figure 31. From a user input desired range and muzzle velocity, the QE Finder procedure searches for the required QE to reach that range. With the outputs QE and muzzle velocity, the trajectory outputs are calculated with the IFDAP trajectory model. The outputs of the trajectory model from the user specified nominal inputs are the nominal trajectory outputs. Next, the perturbations shown in Table 8 are applied to nominal inputs and generated as new inputs for the trajectory model. The trajectory results for those perturbations are then re-computed. These results are used to calculate the ballistic partials necessary to calculate the MPI and Precision accuracies using the models described in Chapter IV. From the MPI and Precision Error metrics, the standard deviation in total range and total deflection, and the overall CEP are calculated using Equations 58, 59, and 61.
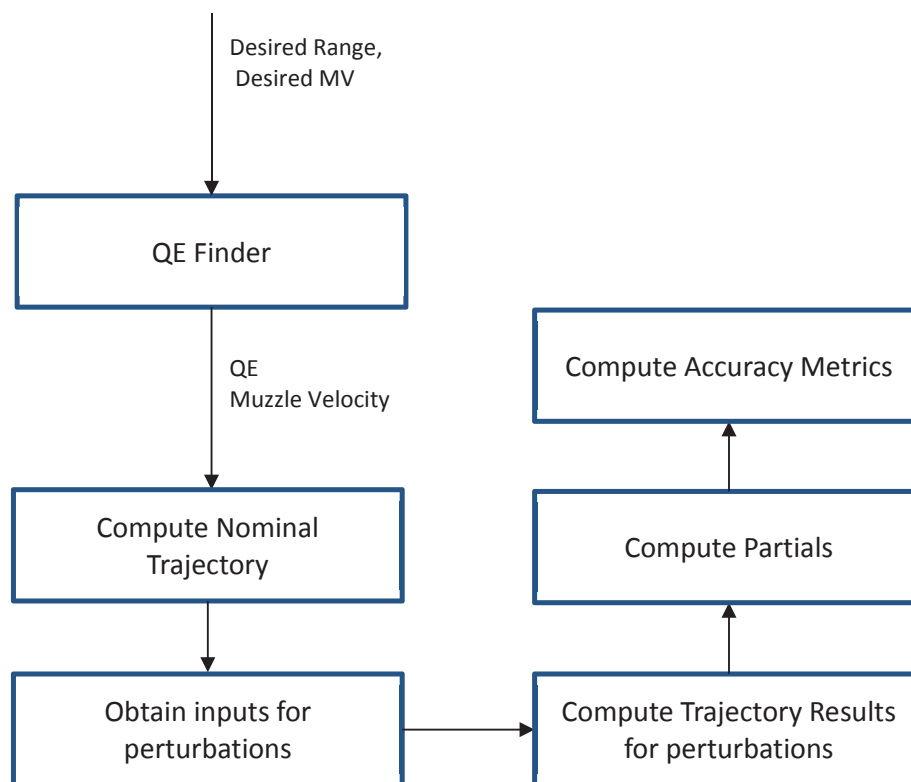
Figure 31. Trajectory, Ballistic Partials, and Accuracy Computation Flow

D.   RESULTS

With the inputs of desired range, muzzle velocity, aerodynamic coefficients, and general parameters from Table 11 and Table 12, the IFDAP generates the following outputs from trajectory and error computations shown in Table 13. In answering the question first asked in opening of this chapter, the standard deviation in range due to precision errors is 46.98 m and due to MPI errors is 112.47 m. The standard deviation in deflection due to precision errors is 14.78 m and due to MPI errors is 38.26 m. This equates to a total standard deviation in range of 121.89 m and total deflection of 41.01 m, with a resulting CEP of 95.9 m.

Table 13. Trajectory and Error Computation Outputs from IFDAP for Case Study

Trajectory Outputs		Error Computation Outputs (m)	
QE	463.48 mils	$\sigma_{x,Prec}$	46.98
Range	15000.09 m	$\sigma_{z,Prec}$	14.78
Initial Velocity	684.00 m/s	$\sigma_{x,MPI}$	112.47
Impact Velocity	314.84 m/s	$\sigma_{z,MPI}$	38.26
Impact Angle	715.51 mils	$\sigma_{x,TOT}$	121.89
Drift(m)	234.11 m	$\sigma_{z,TOT}$	41.01
Drift(mils)	15.61 mils	CEP	95.90
Max Ordinate	2540.68 m		
Range at Max Ord	8474.57 m		
Time of Flight	43.83 s		

E.   IMPLEMENTATION IN VISUAL BASIC

Following its successful coding in MATLAB, the IFDAP was subsequently coded in Microsoft Excel Visual Basic for distribution purposes. This is because most military users do not have MATLAB software, whereas Microsoft Excel is more readily available. Furthermore, various end users with different purposes may take

specific modules of the Visual Basic code and use them in their original state or modify them to suit the users' specific needs. The IFDAP in the Visual Basic environment is coded in the following modules shown in Figure 32. The user interface of the overall model is shown in Figure 33 and the precision and MPI error computation spreadsheets are shown in Figure 34 and Figure 35. The complete Visual Basic code is included in Appendix D through Appendix J.
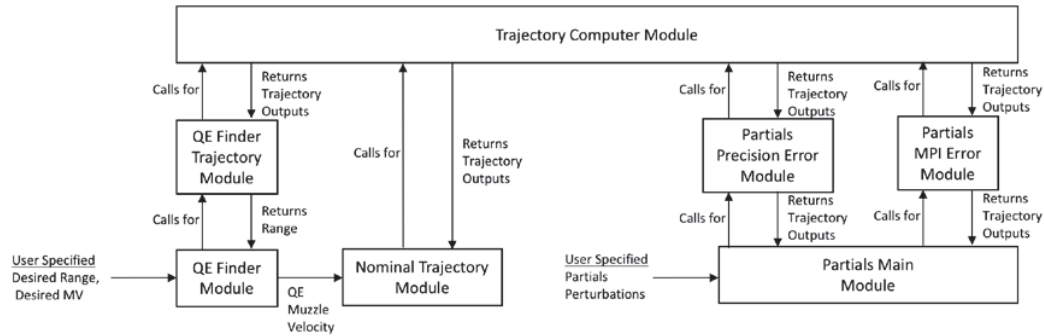
Figure 32. Computation Flow in Visual Basic Environment

Trajectory Computer: The trajectory computer module is the heart of the IFDAP trajectory model. It takes all the inputs gathered from the Nominal Trajectory and QE Finder trajectory modules and computes the output of range, impact velocity, impact angle, drift in mils and meters, max ordinate, and time of flight. This module implements the algorithms described in Chapter III Section C.

QE Finder: This code finds the QE required for the desired range input by the user. First, it obtains the QE for a zero drag trajectory scenario, then inputs this QE into the QE Finder trajectory module as an input for the trajectory computer module to calculate the range. The value of the difference in the range divided by the unit effect of QE is added to the zero drag QE for a second iteration. This goes on until the desired accuracy has been met or the number of iterations has been fulfilled. This module implements the algorithms described in Section B of this chapter.

QE Finder Trajectory: This is an input gathering module for the QE Finder. These inputs are fed into the Trajectory Computer Module.

Nominal Trajectory: This is an input gathering module for a trajectory computation with the inputs specified by the user in the main graphical user interface (GUI) spreadsheet. This module calls the Trajectory Computer module to calculate the outputs and writes the outputs in the Nominal Trajectory Outputs section.

Partials Main: This is a main module that calls for two other modules for the computation of perturbed trajectory outputs for MPI Error and Precision Error computations.

Partials Precision Error: This module gathers the relevant perturbations specified by the user in the main GUI spreadsheet necessary to compute ballistic partials in the precision error model and calculates the inputs for the Trajectory Computer. The outputs are written into the Precision Error spreadsheet. This module implements the algorithms described in Chapter IV Section B Subsection 1.

Partials MPI Error Predicted: This module gathers the relevant perturbations specified by the user in the main GUI spreadsheet necessary to compute ballistic partials in the MPI error model and calculates the inputs for the Trajectory Computer. The outputs are written into the Precision Error spreadsheet. This module implements the algorithms described in Chapter IV Section B Subsection 2.

Nominal Trajectory Inputs

Firing Inputs
Initial Velocity	684	m/s
QE	463.48	mils
Launch Altitude	0	m
Height of target	0	m
Method of Fire	Predicted	

General Parameters
Time Step	0.01	s
g	9.81	m/s²
Air Cp	1005	J/Kg.K
Air Cv	718	J/Kg.K

Wind Speed
Range (+ for tail)	0	m/s
Cross (+from left)	0	m/s

Command Buttons
Find QE & Compute Norminal Traj | Compute Nominal Trajectory | Compute Accuracy

QE Finder Inputs
Desired Range	15000	m
Desired Initial Velocity	684	m/s
Number of iterations	20	
Range Accuracy	1	m

Nominal Trajectory Outputs
QE	463.48	mils
Range	15000.09	m
Initial Velocity	684.00	m/s
Impact Velocity	314.84	m/s
Impact Angle	715.51	mils
Drift(m)	234.11	m
Drift(mils)	15.61	mils
Max Ordinate	2540.68	m
Range at Max Ord	8474.57	m
Time of Flight	43.83	s

Computation Indicator
Computing | Computation Completed

QE Finder Outputs
QE	463.48	mils
Caluclated Range	15000.09	

User Notes

General Usage

1. Using the QE finder, input desired range and MV to find QE required to reach the range. Clicking "Find QE and Compute Nominal Traj" activates procedure to search QE and automatically writes the found QE and desired MV into trajectory model firing inputs and compute nominal trajectory outputs.

2. Clicking "Compute Nomial Trajectory" will activate the procedure to run the trajectory model at conditions specified under "Nominal Trajectory Inputs"

3. Clicking "Compute Accuracy" will activate the procedure to run the trajectory model at perturbed conditions to obtain results of all perturbations required to compute ballistic partials

QE Finder

The QE finder will use the QE unit effect to estimate the next iteration QE. The procedure will run for the specified number of iterations to get within the range accuracy specified. If accuracy is met, procedure will stop, otherwise it will stop at the number of iterations specified.

Error Computation Inputs

Error Budgets (For Precision Error)
σ_v	0.92
σ_drag	0.65
σ_θ	0.3
a_0	0.52
a_1	2000

For MPI Error
Met Staleness | 0.5 hour, 10KM

Error Budgets
σ_lift	1
σ_v	3
σ_drag	1
σ_θ	0.5
σ_azimuth	1
σ_wind	2.83
σ_p	0.27
σ_T	0.3
σ_Chart-X	0
σ_Chart-Z	0
σ_Loc-X	15
σ_Loc-Z	15

Error Computation Outputs

For Precision Error
Ballistic Partials		σ_range	σ_deflection
dX/dV	23.74	21.84	---
dX/drho	-63.64	-41.37	---
dX/dQE	14.38	4.31	---
σ_TOT		46.98	14.78

For MPI Error
Ballistic Partials		σ_range	σ_deflection
dZ/d(lift)	2.30	---	2.30
dX/dV	23.74	71.23	---
dX/drho	-63.64	-63.64	---
dX/dQE	14.38	7.19	---
dZ/da	14.73	---	14.73
dX/dW	19.17	54.26	---
dZ/dW	11.27	---	31.88
dX/drho	-63.64	-17.18	---
dX/dT	10.48	3.15	---
dX/dChart-X	1	0	---
dX/dChart-Z	1	---	0
dX/dLoc-X	1	15	---
dX/dLoc-Z	1	---	15
σ_TOT		112.47	38.26

Partials Perturbations (increase)
Tail Wind (knots)	50
Cross Wind (knots)	50
Air Temp (%)	10
Air Density (%)	10
QE (mils)	50
Initial Velocity (m/s)	10

Delivery Accuracy
σ_X	121.89
σ_Z	41.01
CEP_50	95.90

Figure 33. Main User Interface of IFDAP

MPI Error Calculator

met staleness hour: .5 hour, 10KM

σ_v	σ_{drag}	σ_{theta}	$\sigma_{azimuth}$	σ_{wind}	σ_p	σ_T	$\sigma_{Chart-X}$	$\sigma_{Chart-Z}$	σ_{Loc-X}	σ_{Loc-Z}	σ_{lift}
3	1	0.5	1	2.83	0.27	0.3	0	0	15	15	1

	QE (mils)	Range (m)	Initial Velocity (m/s)	Impact Velocity (m/s)	Impact Angle (mils)	Drift (m)	Drift (mils)	Max Ord (m)	Range at Max Ord (m)	TOF (s)
Nominal	463.48	15000.09	684.00	314.84	715.51	234.11	15.61	2540.68	8474.57	43.83
Perturbation in Tail Wind	463.48	15958.69	684.00	334.70	682.26	252.93	15.85	2614.74	8848.84	44.57
Difference (Perturbation - Nominal)	0.00	958.60	0.00	19.86	-33.25	18.82	0.24	74.06	374.27	0.74
Perturbation in Cross Wind	463.48	15001.23	684.00	314.87	715.43	797.41	53.16	2540.76	8474.81	43.83
Difference (Perturbation - Nominal)	0.00	1.14	0.00	0.04	-0.08	563.30	37.55	0.08	0.24	0.00
Perturbation in Air Temperature	463.48	15104.93	684.00	323.78	696.27	229.88	15.22	2519.25	8430.35	43.61
Difference (Perturbation - Nominal)	0.00	104.84	0.00	8.94	-19.24	-4.23	-0.39	-21.43	-44.22	-0.22

Partials

dX/dw	dZ/dw	dX/dT	dZ/dT	dZ/d(lift)	dZ/da
19.17	11.27	10.48	14.73	2.30	14.73
σ_{X-W}	σ_{Z-W}	σ_{X-T}	σ_{Z-lift}	σ_{Z-azi}	
54.26	31.88	3.15	2.30	14.73	
σ_{MPI-X}	σ_{MPI-Z}				
112.47	38.26				

Figure 34. MPI Error Computation GUI

Precision Error Calculator

					σ_{drag}	σ_v	σ_θ		a0	a1
					0.65	0.92	0.3		0.52	2000
	QE (mils)	Range (m)	Initial Velocity (m/s)	Impact Velocity (m/s)	Impact Angle (mils)	Drift (m)	Drift (mils)	Max Ord (m)	Range at Max Ord (m)	TOF (s)
Nominal	463.48	15000.09	684.00	314.84	715.51	234.11	15.61	2540.68	8474.57	463.48
Perturbation in Initial Velocity	463.48	15237.51	694.00	315.30	721.15	240.18	15.76	2594.86	8634.89	44.28
Difference (Perturbation - Nominal)	0.00	237.42	10.00	0.47	5.63	6.06	0.15	54.18	160.32	-419.20
Perturbation in Air Density	463.48	14363.68	684.00	310.58	713.48	225.46	15.70	2441.66	8086.07	42.83
Difference (Perturbation - Nominal)	0.00	-636.41	0.00	-4.25	-2.03	-8.65	0.09	-99.02	-388.50	-420.65
Perturbation in QE	513.48	15719.13	684.00	317.34	769.82	275.11	17.50	2982.09	8812.00	47.51
Difference (Perturbation - Nominal)	50.00	719.04	0.00	2.50	54.31	40.99	1.89	441.41	337.43	-415.97

Accuracy

	$\sigma_{X\text{-}p}$	$\sigma_{X\text{-}QE}$
	4.31	

dX/dv	dX/dp	dX/dQE
23.74	-63.64	14.38

$\sigma_{X\text{-}v}$	$\sigma_{X\text{-}p}$	$\sigma_{X\text{-}QE}$
21.84	-41.37	4.31

$\sigma_{PE\text{-}X}$	$\sigma_{PE\text{-}Z}$
46.98	14.78

Figure 35. Precision Error Computation GUI

VI. MODEL VALIDATION

A. COMPARATIVE STUDY CASES SELECTION

In Chapter V, one case study demonstrated the use of IFDAP and the outputs it can generate. In this chapter, the IFDAP model is validated against the FT 155-AM-02 firing table for five different muzzle velocities with three ranges each representing short range, medium range, and long range firings. The results are compared and discussed in the next chapter. Since there are no public documents stating accuracies, accuracy values are unable to be compared directly. However, we can calculate the accuracy metrics using the ballistic partials from the firing table and compare them to those generated from IFDAP using the same set of error budgets. The IFDAP developed in the Visual Basic environment is used to generate the trajectory and ballistic partials results for this comparative study.

The following cases are selected from FT 155-AM-02 for the comparative study using the 155mm M107(HE) Projectile (Table 14).

Table 14. Comparative Study Test Cases on 155mm M107(HE) Projectile

Charge	Muzzle Velocity (m/s)	Selected Ranges (m)
4W	337	2000, 3000, 4000
5W	397	3000, 6000, 8000
6W	474	7000, 9000, 11000
7W	568	7000, 10000, 13000
8W	684	8000, 11000, 16000

B. TRAJECTORY RESULTS

For the selected ranges in Table 14, the first step was to use the QE Finder to generate the required QE to reach the desired range. This QE may differ from the FT 155-AM-02 by a small amount. Thus, the trajectory comparison is done at fixed ranges, and the values of QE are compared. The trajectory model of the IFDAP is

subsequently used to compute the trajectory outputs and ballistic partials. The trajectory results for various muzzle velocities are shown in Table 15 through Table 19.

Table 15. Trajectory Results Comparison for Muzzle Velocity of 337 m/s

	Outputs	2000 m		3000 m		4000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 337 m/s (Charge 4)	QE (mils)	96.79	97.2	150.93	152	209.55	211.6
	Range (m)	2000.56	2000	2999.45	3000	4000.67	4000
	Impact Velocity (m/s)	303.22	300	293.83	290	285.90	281
	Impact Angle (mils)	103.50	105	164.31	167	231.42	236
	Drift(m)	3.38	3.2	7.97	7.8	15.25	15.2
	Drift(mils)	1.69	1.6	2.66	2.6	3.81	3.8
	Max Ordinate (m)	49.37	50	117.20	119	220.53	224
	Time of Flight (s)	6.34	6.4	9.76	9.8	13.38	13.5

Table 16. Trajectory Results Comparison for Muzzle Velocity of 397 m/s

		3000 m		6000 m		8000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 397 m/s (Charge 5)	QE (mils)	118.00	118.1	279.28	280.4	417.05	420.6
	Range (m)	2999.81	3000	5998.93	6000	8000.58	8000
	Impact Velocity (m/s)	311.54	310	289.41	287	282.95	280
	Impact Angle (mils)	138.72	140	334.48	338	497.88	506
	Drift(m)	7.80	7.5	36.08	36.6	77.17	79.2
	Drift(mils)	2.60	2.5	6.01	6.1	9.65	9.9
	Max Ordinate (m)	95.55	96	473.33	477	975.94	988
	Time of Flight (s)	8.78	8.8	19.43	19.6	27.95	28.2

Table 17. Trajectory Results Comparison for Muzzle Velocity of 474 m/s

		7000 m		9000 m		11000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 474 m/s (Charge 6)	QE (mils)	258.48	258.4	377.60	378.6	534.79	539.9
	Range (m)	6999.67	7000	8999.67	9000	10999.28	11000
	Impact Velocity (m/s)	299.03	298	294.51	293	296.92	294
	Impact Angle (mils)	346.05	347	493.60	497	671.81	681
	Drift(m)	45.19	45.5	86.61	88.2	163.11	169.4
	Drift(mils)	6.46	6.5	9.62	9.8	14.83	15.4
	Max Ordinate (m)	546.65	548	1054.00	1059	1897.70	1924
	Time of Flight (s)	20.71	20.8	28.70	28.9	38.63	39.1

Table 18. Trajectory Results Comparison for Muzzle Velocity of 568 m/s

		7000 m		10000 m		13000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 568 m/s (Charge 7)	QE (mils)	178.01	177.6	319.81	319.8	518.13	520.7
	Range (m)	6999.25	7000	9998.69	10000	12999.52	13000
	Impact Velocity (m/s)	314.33	313	302.80	302	309.17	307
	Impact Angle (mils)	267.43	269	474.68	477	707.60	713
	Drift(m)	36.81	37.1	92.58	94	205.64	211.9
	Drift(mils)	5.26	5.3	9.26	9.4	15.82	16.3
	Max Ordinate (m)	383.02	383	1052.85	1055	2318.26	2335
	Time of Flight (s)	17.53	17.6	28.49	28.6	42.21	42.6

Table 19. Trajectory Results Comparison for Muzzle Velocity of 684 m/s

		8000 m		11000 m		16000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 684 m/s (Charge 8)	QE (mils)	143.52	141.6	251.98	248.4	534.82	525.3
	Range (m)	7999.42	8000	10998.27	11000	16000.49	16000
	Impact Velocity (m/s)	337.44	338	309.21	309	318.38	318
	Impact Angle (mils)	229.83	228	430.65	427	791.87	785
	Drift(m)	38.23	37.6	92.48	92.4	293.60	292.8
	Drift(mils)	4.78	4.7	8.41	8.4	18.35	18.3
	Max Ordinate (m)	357.04	352	943.53	930	3176.64	3116
	Time of Flight (s)	17.03	17	27.18	27.2	49.06	48.9

C.   BALLISTIC PARTIALS RESULTS

The ballistic partials results are shown in Table 20 through Table 24.

Table 20. Ballistic Partials Comparison for Muzzle Velocity of 337 m/s

	Outputs	2000 m		3000 m		4000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 337 m/s (Charge 4)	dx/dV	7.80	7.50	10.18	9.80	12.49	12.00
	dx/dρ	-1.47	-1.4	-2.55	-2.7	-4.15	-4.4
	dx/dQE	18.49	19.00	17.17	18.00	15.81	16.00
	dx/dW	1.69	1.70	3.15	3.10	4.67	4.50
	dz/dW	0.21	0.22	0.44	0.45	0.77	0.72
	dx/dT	4.58	4.80	8.11	8.50	11.92	12.00

Table 21.   Ballistic Partials Comparison for Muzzle Velocity of 397 m/s

		3000 m		6000 m		8000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 397 m/s (Charge 5)	dx/dV	11.71	11.40	16.82	16.20	19.40	18.90
	dx/dρ	-4.59	-4.6	-10.58	-11.1	-16.58	-17.3
	dx/dQE	20.30	21.00	15.72	16.00	11.89	12.00
	dx/dW	2.35	2.20	8.20	7.70	12.40	11.60
	dz/dW	0.63	0.63	2.17	2.04	3.50	3.28
	dx/dT	3.57	3.70	17.65	17.30	25.89	25.40

Table 22.   Ballistic Partials Comparison for Muzzle Velocity of 474 m/s

		7000 m		9000 m		11000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 474 m/s (Charge 6)	dx/dV	18.46	18.40	20.52	20.40	22.74	22.40
	dx/dρ	-18.10	-18.4	-24.08	-24.9	-31.98	-33.8
	dx/dQE	17.80	18.00	14.33	15.00	9.37	10.00
	dx/dW	6.80	6.40	11.10	10.40	15.50	14.50
	dz/dW	3.06	2.94	4.80	4.50	6.80	6.38
	dx/dT	8.76	8.90	17.98	17.70	25.33	25.00

Table 23.   Ballistic Partials Comparison for Muzzle Velocity of 568 m/s

		7000 m		10000 m		13000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 568 m/s (Charge 7)	dx/dV	17.54	17.40	21.01	20.70	23.70	23.30
	dx/dρ	-23.04	-24.1	-34.09	-35.3	-46.35	-49
	dx/dQE	23.31	25.00	17.44	18.00	11.20	12.00
	dx/dW	4.20	4.00	9.70	9.10	16.30	15.30
	dz/dW	2.68	2.66	5.60	5.30	9.15	8.58
	dx/dT	-2.65	-2.50	7.48	7.60	19.34	19.40

Table 24.   Ballistic Partials Comparison for Muzzle Velocity of 684 m/s

		8000 m		11000 m		16000 m	
		IFDAP	FT	IFDAP	FT	IFDAP	FT
Muzzle Velocity = 684 m/s (Charge 8)	dx/dV	16.91	16.70	20.54	20.20	24.45	24.50
	dx/dρ	-29.59	-31.6	-46.02	-48.2	-68.45	-72.5
	dx/dQE	32.00	34.00	22.40	24.00	12.20	13.00
	dx/dW	3.95	3.70	8.20	7.70	18.50	17.30
	dz/dW	2.74	2.72	5.71	5.50	12.30	11.52
	dx/dT	-7.37	-7.30	-4.11	-4.30	14.04	13.40

D. ACCCURACY RESULTS

The error budgets used in the generation of the accuracy metrics are those from Table 9 and Table 10. A MET staleness of 0.5 hour was assumed, and the corresponding MET error budgets are $\sigma_{wind} = 3.91$, $\sigma_{\rho} = 0.85$, and $\sigma_T = 0.69$. The equations presented in Chapter IV are used to calculate the results of the accuracy metrics shown in Table 25 through Table 29.

Table 25.   Accuracy Comparison for Muzzle Velocity of 337 m/s

		2000 m			3000 m			4000 m		
		IFDAP	FT	% Diff	IFDAP	FT	% Diff	IFDAP	FT	% Diff
Muzzle Velocity = 337 m/s (Charge 4)	$\sigma_{x,Prec}$	9.12	9.00	1.38%	10.82	10.65	1.55%	12.72	12.37	2.80%
	$\sigma_{z,Prec}$	1.59	1.59	0.00%	2.46	2.46	0.00%	3.38	3.38	0.00%
	$\sigma_{x,MPI}$	30.25	29.67	1.97%	37.76	36.94	2.23%	46.06	44.68	3.08%
	$\sigma_{z,MPI}$	15.15	15.15	-0.02%	15.38	15.38	-0.02%	15.80	15.76	0.23%
	$\sigma_{x,TOT}$	31.60	31.00	1.94%	39.28	38.44	2.18%	47.78	46.36	3.05%
	$\sigma_{z,TOT}$	15.23	15.24	-0.04%	15.58	15.58	0.01%	16.16	16.12	0.23%
	CEP	27.57	27.22	1.29%	32.30	31.80	1.57%	37.64	36.79	2.32%

Table 26.   Accuracy Comparison for Muzzle Velocity of 397 m/s

		3000 m			6000 m			8000 m		
		IFDAP	FT	% Diff	IFDAP	FT	% Diff	IFDAP	FT	% Diff
Muzzle Velocity = 397 m/s (Charge 5)	$\sigma_{x,Prec}$	12.73	12.59	1.07%	17.58	17.24	1.97%	21.15	21.02	0.63%
	$\sigma_{z,Prec}$	2.41	2.41	0.00%	5.28	5.28	0.00%	7.65	7.65	0.00%
	$\sigma_{x,MPI}$	41.13	40.27	2.13%	60.17	62.55	-3.80%	72.71	79.72	-8.80%
	$\sigma_{z,MPI}$	15.48	15.48	0.00%	17.17	16.92	1.45%	19.96	19.38	2.99%
	$\sigma_{x,TOT}$	43.05	42.20	2.02%	62.68	64.88	-3.39%	75.72	82.45	-8.16%
	$\sigma_{z,TOT}$	15.67	15.67	0.02%	17.96	17.73	1.30%	21.38	20.84	2.61%
	CEP	34.57	34.06	1.49%	47.48	48.63	-2.37%	57.16	60.80	-5.99%

Table 27.   Accuracy Comparison for Muzzle Velocity of 474 m/s

		7000 m			9000 m			11000 m		
		IFDAP	FT	% Diff	IFDAP	FT	% Diff	IFDAP	FT	% Diff
Muzzle Velocity = 474 m/s (Charge 6)	$\sigma_{x,Prec}$	21.34	21.42	-0.37%	24.89	25.19	-1.18%	29.63	30.27	-2.12%
	$\sigma_{z,Prec}$	6.08	6.08	0.00%	8.40	8.40	0.00%	11.36	11.36	0.00%
	$\sigma_{x,MPI}$	65.18	67.82	-3.90%	76.82	83.05	-7.50%	91.30	101.22	-9.80%
	$\sigma_{z,MPI}$	18.40	18.11	1.62%	21.83	20.83	4.80%	28.70	27.19	5.57%
	$\sigma_{x,TOT}$	68.59	71.13	-3.57%	80.75	86.79	-6.96%	95.98	105.65	-9.15%
	$\sigma_{z,TOT}$	19.38	19.10	1.46%	23.39	22.46	4.14%	30.87	29.46	4.77%
	CEP	51.79	53.12	-2.50%	61.31	64.31	-4.67%	74.68	79.54	-6.11%

Table 28. Accuracy Comparison for Muzzle Velocity of 568 m/s

		7000 m			10000 m			13000 m		
		IFDAP	FT	% Diff	IFDAP	FT	% Diff	IFDAP	FT	% Diff
Muzzle Velocity = 568 m/s (Charge 7)	$\sigma_{x,Prec}$	23.10	23.62	-2.20%	29.87	30.30	-1.43%	37.34	38.56	-3.16%
	$\sigma_{z,Prec}$	5.82	5.82	0.00%	9.01	9.01	0.00%	13.28	13.28	0.00%
	$\sigma_{x,MPI}$	63.11	65.99	-4.36%	80.66	87.19	-7.49%	101.92	114.20	-10.75%
	$\sigma_{z,MPI}$	17.98	17.94	0.24%	23.22	22.12	4.99%	33.83	31.46	7.52%
	$\sigma_{x,TOT}$	67.20	70.09	-4.12%	86.01	92.31	-6.82%	108.54	120.53	-9.95%
	$\sigma_{z,TOT}$	18.89	18.86	0.17%	24.90	23.88	4.26%	36.35	34.15	6.44%
	CEP	50.68	52.36	-3.21%	65.30	68.40	-4.53%	85.30	91.06	-6.33%

Table 29. Accuracy Comparison for Muzzle Velocity of 684 m/s

		8000 m			11000 m			16000 m		
		IFDAP	FT	% Diff	IFDAP	FT	% Diff	IFDAP	FT	% Diff
Muzzle Velocity = 684 m/s (Charge 8)	$\sigma_{x,Prec}$	26.36	27.60	-4.51%	36.00	37.13	-3.05%	49.98	52.38	-4.59%
	$\sigma_{z,Prec}$	6.52	6.52	0.00%	9.52	9.52	0.00%	16.53	16.53	0.00%
	$\sigma_{x,MPI}$	65.69	70.56	-6.90%	84.97	94.65	-10.23%	120.32	139.23	-13.58%
	$\sigma_{z,MPI}$	18.44	18.38	0.30%	23.83	23.06	3.33%	43.32	39.91	8.55%
	$\sigma_{x,TOT}$	70.78	75.77	-6.58%	92.28	101.67	-9.24%	130.29	148.76	-12.41%
	$\sigma_{z,TOT}$	19.56	19.51	0.27%	25.66	24.95	2.85%	46.36	43.20	7.33%
	CEP	53.18	56.09	-5.18%	69.43	74.54	-6.86%	104.00	113.00	-7.97%

VII. DISCUSSION

A. TRAJECTORY ANALYSIS

The percentage errors of various trajectory output variables for the comparison between the IFDAP trajectory model and the FT 155-AM-02 firing table are shown in Figure 36 to Figure 41. It is observed that, in general, the percentage errors tend to increase with increasing range for a given charge, and the maximum error occurs at the maximum range of each charge. This is explained by the fact that the decoupled and simple equations used to model the trajectory have a finite error at each computation step and are compounded with an increasing number of time steps. Hence, for firings with higher TOF, the errors observed are larger. Despite this, maximum errors in trajectory outputs are kept to within 2 percent, with the exception of certain drift calculations that are explained later. Noting that conventional artillery is used for mass effects and not for missions where precise delivery is needed, the accuracy of the trajectory model is sufficient for most applications.

It is also observed that the high percentage errors in drift for short range firings is due to the comparison of small values resulting in high percentage errors. For example, for a short range firing with a muzzle velocity of 337m/s, the actual value of drift from FT 155-AM-02 for this case was 1.6 mils, whereas the IFDAP trajectory model calculated 1.69 mils. At a firing range of 2000 m, this difference is approximately only 0.18m, but 5.76 percent. Higher values of drift errors of up to 8 percent are also observed at the final 1000 m of the maximum range for each charge. However, for most real world applications, a firing charge that would barely reach the desired range would not be chosen because of higher variability in the behavior of the projectile. For typical values of ranges and muzzle velocities, represented by values corresponding to medium range, the drift errors are kept to within 4 percent.
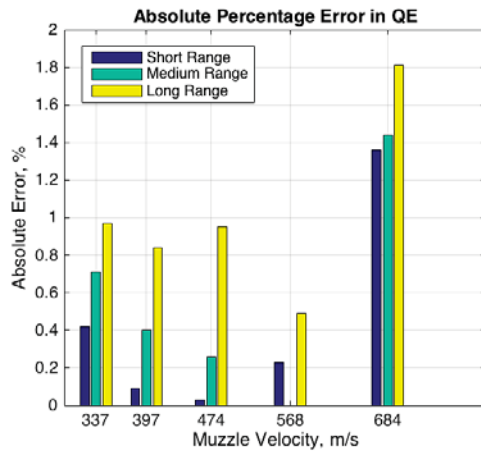
Figure 36. Absolute Percentage
Error in QE
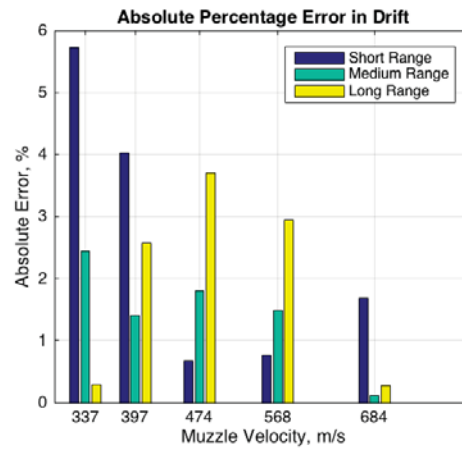
Figure 37. Absolute Percentage
Error in Drift
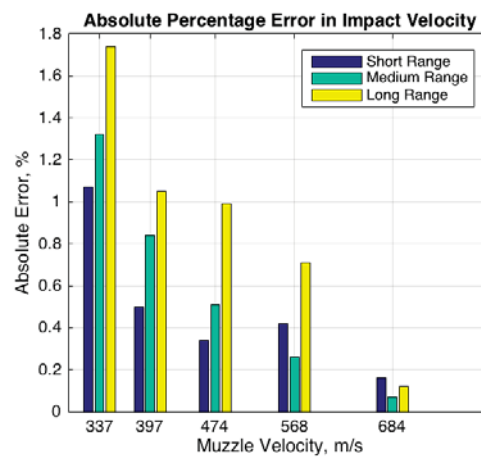
Figure 38. Absolute Percentage
Error in Impact
Velocity
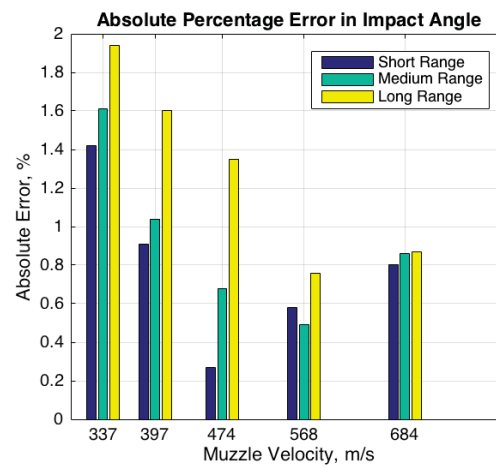
Figure 39. Absolute Percentage
Error in Impact
Angle

70
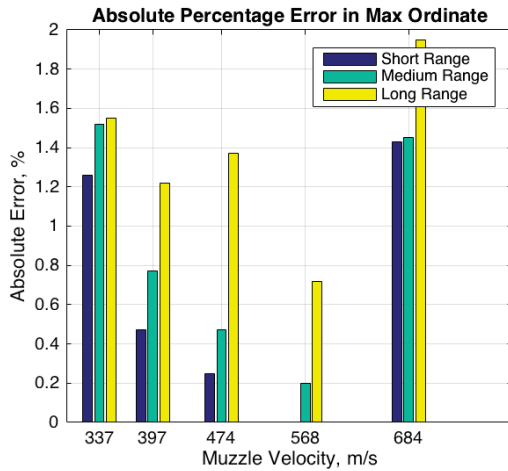
Figure 40. Absolute Percentage Error in Max Ordinate
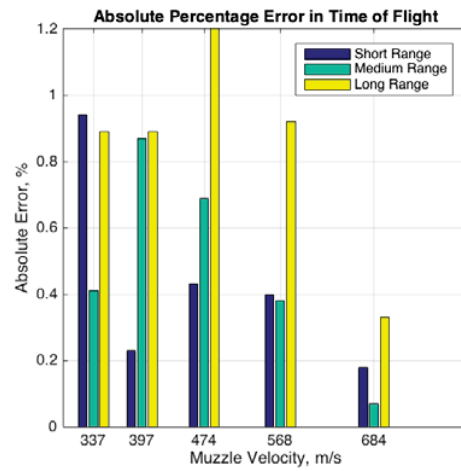
Figure 41. Absolute Percentage Error in Time of Flight

B.   BALLISTIC PARTIALS ANALYSIS

Unlike the trajectory model, errors in the ballistic partials do not have a similar trend although the ballistic partials are derived from the trajectory model. This is because the ballistic partials measure the gradient at the range specified by the perturbations input, and if the two trajectory models are not identical in the equations used, the gradient at specific points may have greater errors than the actual value of the variable, especially for small value variables such as the wind partials. In Table 30, supported by Figure 42, an example is shown using QE as the variable of interest. Between 8000 m and 9000 m, the gradient of the IFDAP line 1000/(175.5-143.5) = 31.25 and for the FT line is 1000/(172-142) = 32.89. This corresponds to a percentage error in the dx/dQE gradient of (31.25-32.89)/32.89 = 5%, although the percentage error in QE at 8000 m is only (143.5-141.6)/141.6 = 1.34%, and (175.23-172.4)/172.4 = 1.64% at 9000 m.

Table 30. Absolute % Error in QE vs. Absolute % Error in dx/dQE

MV (m/s)	Average Absolute % error in QE	Max Absolute % error in QE	Average Absolute % error in dx/dQE	Max Absolute % error in dx/dQE
337	0.70%	0.97%	2.82%	4.63%
397	0.44%	0.84%	2.02%	3.36%
474	0.39%	0.95%	3.95%	6.30%
568	0.09%	0.49%	5.51%	6.77%
684	1.54%	1.81%	6.23%	6.67%
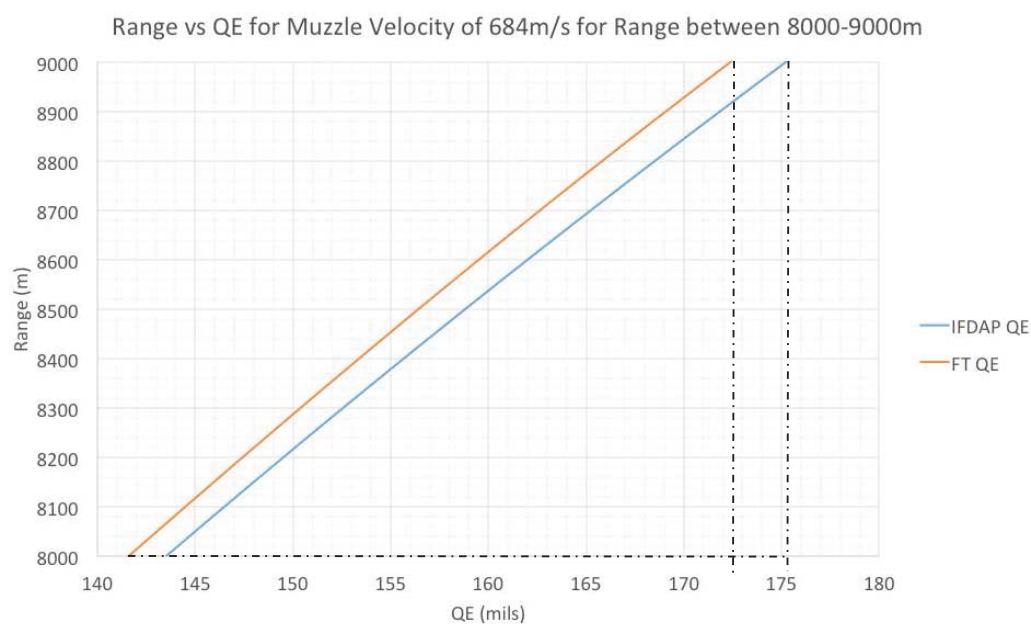
Figure 42. Difference in Gradient Causes Higher Percentage Error in Ballistic Partials Compared to the Corresponding Trajectory Variable

The average and maximum errors for all ballistic partials of interest in the error models are shown in Table 31. The wind partials have the highest average and maximum percentage error due to the small magnitudes of the variables. The maximum error of all partials is less than 7 percent.

Table 31.   Average and Maximum Percentage Errors for Ballistic Partials

Ballistic Partials	Average Error	Maximum Error
dx/dV	2.05%	4.07%
dx/dρ	4.34%	6.37%
dx/dQE	4.11%	6.77%
dx/dW	5.62%	6.94%
dz/dW	4.62%	6.94%
dx/dT	2.64%	6.00%

C.   ACCURACY ANALYSIS

Similar to the trend seen in the trajectory model, the percentage error in the accuracy metrics increases with increasing range and muzzle velocities. From Figure 43 and Figure 44, the absolute percentage error in range is less than 13 percent and in deflection, less than 8 percent. For typical firing range, represented by values corresponding to medium range, the absolute error in range is less than 10 percent and in deflection, less than 5 percent. The standard deviation in range, $\sigma_{x,tot}$, is observed to be greater in magnitude than the corresponding deflection. This is because more partials are required to compute $\sigma_{x,tot}$, and each partial's error increases the total magnitude, resulting in a larger error.
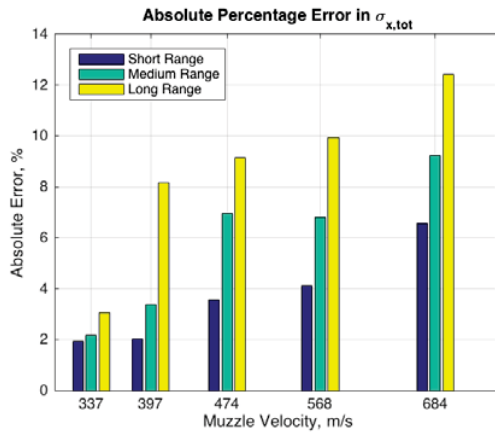
Figure 43. Absolute Percentage Error in $\sigma_{x,tot}$
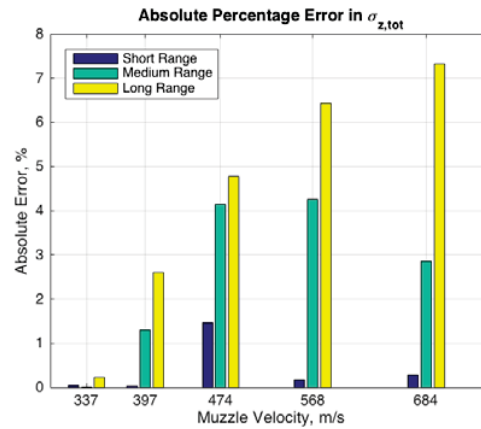
Figure 44. Absolute Percentage Error in $\sigma_{z,tot}$

Note that not all partials have equal influence on the overall error metric. It depends on the relative magnitude of the partials and error budgets. A large partial with a significantly smaller error budget may have a small effect as compared to a small partial with a large error budget. The value of the error budgets depends on the weapon platform, firing charge, projectile, and MET staleness, and the partials depend on the firing weapon, firing charge, and projectile. In the case of M198 firing M107(HE) 155mm, it is observed that the magnitude of the partials changes with range and muzzle velocity as shown in Figure 45. At short range and low muzzle velocity, the largest two partials in magnitude are dx/dQE and dx/dV. The value of dx/dV increases in magnitude with increasing range and eventually becomes larger than dx/dQE. At increasing muzzle velocities, the value of dx/dρ becomes more dominant. At short range and high muzzle velocity, dx/dQE and dx/dρ are most dominant. Similar to lower muzzle velocities, dx/dV increases with increasing range and surpasses dx/dQE, but dx/dρ increases more than the other two partials and becomes most dominant at long range and high muzzle velocity firings.
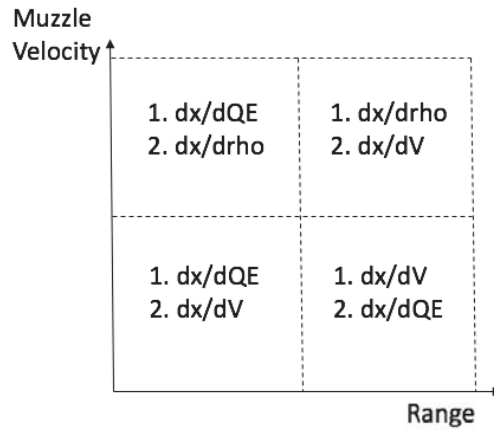
Figure 45. Top Two Largest Partials for Various Muzzle Velocity and Range Combinations

D. EFFECT OF TIME STEP

The magnitude of the time step has an effect on the error observed. For all illustrations in this thesis, a time step of 0.01 seconds is chosen. This selection is based on an initial observation that the trajectory outputs had no significant difference with smaller time steps. As the magnitude of time step is reduced, the trajectory outputs and ballistic partials converge toward those in FT 155-AM-02. However, this is at the expense of computational speed, as shown in Table 32. If the accuracy metrics are computed using smaller time steps, the error is less than those found in Table 25 to Table 29.

Table 32. Effect of Time Step on QE and dx/dQE for Muzzle Velocity of
684 m/s at 11,000 m

	dt = 0.1 sec	dt = 0.01 sec	dt = 0.005 sec	FT
QE (mils)	254.57	251.98	251.84	248.4
% error in QE	2.49%	1.44%	1.39%	---
dx/dQE	19.88	20.50	20.26	20.20
% error in dx/dQE	-1.59%	1.50%	0.30%	---
Computation Time	3secs	30secs	80 secs	---

VIII. CONCLUSIONS

This research successfully developed a program to quickly estimate the accuracy of unguided artillery projectiles through three intermediate steps:

1. Developing a simplified version of the modified point mass trajectory model,

2. Calculating ballistic partials using the developed trajectory model, and

3. Evaluating error metrics using well known error models for MPI and Precision Errors, and the appropriate error budgets associated with the method of fire and MET staleness.

The following important observations were made during the work of this thesis:

1. Accuracy computation can typically be achieved in under 30 seconds for most ranges and charges.

2. The smaller the magnitude of time step, the better the convergence of the trajectory and ballistic partials outputs.

3. The angle of attack is sufficiently small to be neglected.

4. The errors are the highest at the minimum and maximum range of a given charge.

5. By comparing with data from FT 155-AM-02 firing tables, maximum error of trajectory outputs, except drift, are less than 2 percent and less than 4 percent for drift. Ballistic partials are less than 7 percent.

6. Using a common set of error budgets to compute the accuracy metrics using partials from FT 155-AM-02 and comparing them with those generated from IFDAP, the maximum error in the accuracy metric $\sigma_{x,tot}$ is less than 13 percent, and $\sigma_{z,tot}$ is less than 8 percent. For medium (typical) firing ranges, $\sigma_{x,tot}$ is less than 10 percent, and $\sigma_{z,tot}$ is less than 5 percent.

7. Not all ballistic partials have equal weighting on error metrics. The three largest partials numerically are dx/dQE, dx/dV, and dx/dρ. Depending on muzzle velocity and range, one of them would be largest. However, their weighting on accuracy would depend on the corresponding error budgets.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. DRIFT CALCULATIONS

Calibre/spin, L
V_0
Bore diameter, d

For Initial Spin, p_0

$$p_0 = \frac{2\pi v_0}{Ld}$$

drag.m interpolation $\xrightarrow{C_{lp}, \rho}$

V, p

Spin Damping Moment =
$$\frac{1}{2}\rho V^2 Sd(pd/V)C_{lp}$$

$\dot{p} = Spin\ Damping\ Moment\ /\ I_{xx}$

$p(t+dt) = p(t) + \dot{p}(t)\Delta t$

drag.m interpolation $\xrightarrow{C_{m_\alpha}}$

$$\beta = \frac{8pI_{xx}g\cos\theta}{\pi\rho D^2 L_{ref}\, C_{m_\alpha}V^3}$$

$$a_z = \frac{1}{2}S\rho V^2\,\frac{(C_{N_\alpha}\sin\beta\cos(\phi+\beta) - C_d\cos\theta\sin\phi)}{m}$$

$v_z(t+dt) = v_z(t) + a_z(t)\Delta t$

$z(t+dt) = z(t) + v_z(t+1)\Delta t$

p

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. MATLAB SCRIPT FOR POINT MASS, ZERO DRAG MODEL

```matlab
clear all, clc
theta = (456/3200)*pi; %input elevation
v0=684; %input projectile velocity

v0h = v0*cos(theta);
vh(1)=v0h; sh(1)=0;
v0v = v0*sin(theta);
vv(1)=v0v; sv(1)=0;

a=-9.81;
t = linspace(0,99);
[r c] = size(t);

for i=2:c
    vv(i) = v0v+a*t(i);
    sv(i) = v0v*t(i)+0.5*a*t(i)^2;

    vh(i) = v0h;
    sh(i) = v0h*t(i);

    if sv(i)<0;
        break
    end

end

impactvelocity = sqrt(vh(i)^2 + vv(i)^2);
impactangle=atan(abs(vv(i)/vh(i)));
timeofflight=-(2*v0v)/a;
maxaltitude=max(sv);

plot(sh,sv)
xlabel('Range (m)'), ylabel('Altitude (m)');
axis([0 inf 0 inf]), title('Point Mass, Zero Drag Model')

fprintf('Range is %.2f m \n', sh(i))
fprintf('Impact Angle is %.2f radians \n', impactangle)
fprintf('Time of Flight is %.2f s \n', timeofflight)
fprintf('Impact Velocity is %.2f m/s \n', impactvelocity)
fprintf('Max Altitude is %.2f m \n', maxaltitude)
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. MET ERROR BUDGETS

Time Staleness (hr) [0.5 to 12]	Separation (Distance) Staleness (km)
0.5	10

	Density				Wind				Temperature			
Met Line	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)	Instrument Error (kts)	Time Staleness Error (kts)	Distance (Separation and Balloon Drift) Staleness Error (kts)	Total Error (kts)	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)
1	0.15	0.65	0.53	0.85	1.65	1.84	3.03	3.91	0.25	0.50	0.41	0.69
2	0.15	0.41	0.34	0.55	1.04	1.90	1.55	2.66	0.25	0.31	0.26	0.48
3	0.15	0.29	0.24	0.40	0.55	1.99	1.63	2.63	0.25	0.22	0.18	0.38
4	0.15	0.24	0.20	0.34	0.42	2.05	1.69	2.70	0.25	0.18	0.15	0.34
5	0.15	0.21	0.17	0.31	0.41	2.10	1.75	2.76	0.25	0.16	0.13	0.32
6	0.15	0.17	0.14	0.27	0.25	2.14	1.84	2.83	0.25	0.13	0.11	0.30
7	0.15	0.15	0.13	0.25	0.21	2.16	1.97	2.93	0.25	0.11	0.10	0.29
8	0.15	0.13	0.13	0.24	0.17	2.17	2.13	3.05	0.25	0.10	0.10	0.29
9	0.15	0.12	0.13	0.23	0.13	2.21	2.36	3.23	0.25	0.09	0.10	0.28
10	0.15	0.10	0.13	0.22	0.11	2.41	2.96	3.81	0.25	0.08	0.10	0.28
11	0.15	0.09	0.13	0.22	0.07	2.47	3.60	4.36	0.25	0.07	0.10	0.28
12	0.15	0.08	0.14	0.22	0.06	2.42	4.08	4.74	0.25	0.06	0.11	0.28
13	0.15	0.08	0.15	0.22	0.05	2.20	4.11	4.66	0.25	0.06	0.11	0.28
14	0.15	0.07	0.15	0.22	0.04	1.93	3.87	4.32	0.25	0.06	0.11	0.28
15	0.15	0.07	0.14	0.22	0.03	1.65	3.45	3.82	0.25	0.05	0.11	0.28

Time Staleness (hr) [0.5 to 12]	Separation (Distance) Staleness (km)
2	20

	Density				Wind				Temperature			
Met Line	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)	Instrument Error (kts)	Time Staleness Error (kts)	Distance (Separation and Balloon Drift) Staleness Error (kts)	Total Error (kts)	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)
1	0.15	1.30	0.75	1.51	1.65	3.68	3.47	5.32	0.25	0.99	0.57	1.18
2	0.15	0.82	0.48	0.96	1.04	3.80	2.19	4.51	0.25	0.63	0.36	0.77
3	0.15	0.58	0.34	0.69	0.55	3.98	2.30	4.63	0.25	0.44	0.26	0.57
4	0.15	0.47	0.27	0.57	0.42	4.11	2.38	4.76	0.25	0.36	0.21	0.49
5	0.15	0.41	0.24	0.50	0.41	4.19	2.44	4.87	0.25	0.31	0.18	0.44
6	0.15	0.34	0.20	0.42	0.25	4.27	2.50	4.96	0.25	0.26	0.15	0.39
7	0.15	0.29	0.17	0.37	0.21	4.31	2.58	5.03	0.25	0.22	0.13	0.36
8	0.15	0.26	0.16	0.34	0.17	4.33	2.66	5.09	0.25	0.20	0.12	0.34
9	0.15	0.24	0.15	0.32	0.13	4.41	2.81	5.23	0.25	0.18	0.12	0.33
10	0.15	0.21	0.14	0.29	0.11	4.81	3.32	5.85	0.25	0.16	0.11	0.31
11	0.15	0.18	0.14	0.28	0.07	4.93	3.84	6.25	0.25	0.14	0.11	0.31
12	0.15	0.17	0.15	0.27	0.06	4.84	4.24	6.43	0.25	0.13	0.11	0.30
13	0.15	0.16	0.15	0.26	0.05	4.39	4.22	6.09	0.25	0.12	0.11	0.30
14	0.15	0.15	0.15	0.26	0.04	3.85	3.94	5.51	0.25	0.11	0.11	0.30
15	0.15	0.14	0.15	0.25	0.03	3.30	3.51	4.82	0.25	0.10	0.11	0.29

Time Staleness (hr) [0.5 to 12]	Separation (Distance) Staleness (km)	
4	40	

Met Line	Density				Wind				Temperature			
	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)	Instrument Error (kts)	Time Staleness Error (kts)	Distance (Separation and Balloon Drift) Staleness Error (kts)	Total Error (kts)	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)
1	0.15	1.84	1.06	2.13	1.65	5.20	3.70	6.59	0.25	1.41	0.81	1.64
2	0.15	1.16	0.67	1.35	1.04	5.37	3.10	6.29	0.25	0.89	0.51	1.06
3	0.15	0.82	0.48	0.96	0.55	5.63	3.25	6.53	0.25	0.63	0.36	0.77
4	0.15	0.67	0.39	0.79	0.42	5.81	3.36	6.72	0.25	0.51	0.30	0.64
5	0.15	0.58	0.34	0.69	0.41	5.93	3.43	6.86	0.25	0.44	0.26	0.57
6	0.15	0.47	0.28	0.57	0.25	6.04	3.50	6.99	0.25	0.36	0.21	0.49
7	0.15	0.41	0.24	0.50	0.21	6.10	3.55	7.06	0.25	0.31	0.18	0.44
8	0.15	0.37	0.22	0.45	0.17	6.13	3.60	7.11	0.25	0.28	0.17	0.41
9	0.15	0.34	0.20	0.42	0.13	6.24	3.71	7.26	0.25	0.26	0.15	0.39
10	0.15	0.29	0.18	0.37	0.11	6.81	4.16	7.98	0.25	0.22	0.14	0.36
11	0.15	0.26	0.17	0.34	0.07	6.97	4.51	8.31	0.25	0.20	0.13	0.34
12	0.15	0.24	0.16	0.33	0.06	6.84	4.74	8.32	0.25	0.18	0.13	0.33
13	0.15	0.22	0.16	0.31	0.05	6.21	4.58	7.72	0.25	0.17	0.12	0.33
14	0.15	0.21	0.16	0.30	0.04	5.45	4.21	6.89	0.25	0.16	0.12	0.32
15	0.15	0.19	0.15	0.29	0.03	4.66	3.72	5.96	0.25	0.15	0.12	0.31

Time Staleness (hr) [0.5 to 12]	Separation (Distance) Staleness (km)	
4	48	

Met Line	Density				Wind				Temperature			
	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)	Instrument Error (kts)	Time Staleness Error (kts)	Distance (Separation and Balloon Drift) Staleness Error (kts)	Total Error (kts)	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)
1	0.15	1.84	1.16	2.18	1.65	5.20	3.89	6.70	0.25	1.41	0.89	1.68
2	0.15	1.16	0.74	1.38	1.04	5.37	3.40	6.44	0.25	0.89	0.56	1.08
3	0.15	0.82	0.52	0.98	0.55	5.63	3.56	6.69	0.25	0.63	0.40	0.79
4	0.15	0.67	0.43	0.81	0.42	5.81	3.67	6.89	0.25	0.51	0.33	0.66
5	0.15	0.58	0.37	0.70	0.41	5.93	3.75	7.03	0.25	0.44	0.28	0.58
6	0.15	0.47	0.30	0.58	0.25	6.04	3.83	7.16	0.25	0.36	0.23	0.50
7	0.15	0.41	0.26	0.51	0.21	6.10	3.88	7.24	0.25	0.31	0.20	0.45
8	0.15	0.37	0.24	0.46	0.17	6.13	3.92	7.28	0.25	0.28	0.18	0.42
9	0.15	0.34	0.22	0.43	0.13	6.24	4.03	7.43	0.25	0.26	0.17	0.39
10	0.15	0.29	0.19	0.38	0.11	6.81	4.49	8.15	0.25	0.22	0.15	0.37
11	0.15	0.26	0.18	0.35	0.07	6.97	4.80	8.46	0.25	0.20	0.14	0.35
12	0.15	0.24	0.17	0.33	0.06	6.84	4.97	8.46	0.25	0.18	0.13	0.34
13	0.15	0.22	0.17	0.32	0.05	6.21	4.76	7.82	0.25	0.17	0.13	0.33
14	0.15	0.21	0.16	0.30	0.04	5.45	4.35	6.97	0.25	0.16	0.13	0.32
15	0.15	0.19	0.16	0.29	0.03	4.66	3.83	6.03	0.25	0.15	0.12	0.32

Time Staleness (hr) [0.5 to 12]	Separation (Distance) Staleness (km)	
12	48	

Mid Latitude Mean Wind Profile Optimized for Annual Winds (FARMY-153-5) NAVAIDS (MDS) Used for Instrument Error

	Density				Wind				Temperature			
Met Line	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)	Instrument Error (kts)	Time Staleness Error (kts)	Distance (Separation and Balloon Drift) Staleness Error (kts)	Total Error (kts)	Instrument Error (%)	Time Staleness Error (%)	Distance (Separation and Balloon Drift) Staleness Error (%)	Total Error (%)
1	0.15	3.19	1.16	3.40	1.65	9.01	#NUM!	9.15	0.25	2.44	0.89	2.61
2	0.15	2.02	0.74	2.15	1.04	9.30	3.40	9.96	0.25	1.54	0.56	1.66
3	0.15	1.42	0.52	1.52	0.55	9.75	3.56	10.40	0.25	1.09	0.40	1.19
4	0.15	1.16	0.43	1.25	0.42	10.06	3.67	10.72	0.25	0.89	0.33	0.98
5	0.15	1.01	0.37	1.08	0.41	10.27	3.75	10.94	0.25	0.77	0.28	0.86
6	0.15	0.82	0.30	0.89	0.25	10.46	3.83	11.14	0.25	0.63	0.23	0.72
7	0.15	0.71	0.26	0.77	0.21	10.57	3.88	11.26	0.25	0.54	0.20	0.63
8	0.15	0.64	0.24	0.70	0.17	10.62	3.92	11.32	0.25	0.49	0.18	0.58
9	0.15	0.58	0.22	0.64	0.13	10.81	4.03	11.53	0.25	0.44	0.17	0.54
10	0.15	0.50	0.19	0.56	0.11	11.79	4.49	12.62	0.25	0.39	0.15	0.48
11	0.15	0.45	0.18	0.51	0.07	12.08	4.80	13.00	0.25	0.34	0.14	0.45
12	0.15	0.41	0.17	0.47	0.06	11.85	4.97	12.85	0.25	0.31	0.13	0.42
13	0.15	0.38	0.17	0.44	0.05	10.75	4.76	11.76	0.25	0.29	0.13	0.40
14	0.15	0.36	0.16	0.42	0.04	9.43	4.35	10.39	0.25	0.27	0.13	0.39
15	0.15	0.34	0.16	0.40	0.03	8.08	3.83	8.94	0.25	0.26	0.12	0.38

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. VISUAL BASIC CODE (QE FINDER MAIN)

Option Explicit

Dim theta_zero, des_range, des_v0, g, i, iterations, accuracy, QE, calc_range

Sub QEfinderMain()

```
'----------------------------------------------------------------
' This code finds the QE required for the desired range input by the user. First,
' it obtains the QE for a zero drag trajectory scenario, then inputs this QE to the thesis
' model to calculate the range. The value of the difference in the range divided by
' the unit effect of QE is added to the zero drag QE for a 2nd iteration. This goes on
' until the desired accuracy has been met, or number of iterations has been fulfilled.
'----------------------------------------------------------------

'Clearing output cells
range("F27:F28").Select
Selection.clearcontents

' Getting General Parameters
g = range("B9")
des_range = range("B27")
des_v0 = range("B28")


' Calculating QE for zero drag trajectory
theta_zero = 0.5 * WorksheetFunction.Asin(g * des_range / des_v0 ^ 2)
QE = theta_zero / WorksheetFunction.Pi * 3200


' 1st iteration using QE from zero drag
Call QEFinderTrajectory(QE, calc_range)


' Iterating towards desired accuracy
iterations = range("B29")
accuracy = range("B30")

For i = 1 To iterations

    If Abs(des_range - calc_range) > accuracy Then
```

```vba
        QE = QE + (des_range - calc_range) / (calc_range / QE)
        Call QEFinderTrajectory(QE, calc_range)

    End If

Next i

' Writing results to Nominal Trajectory Input Cells
range("F27") = QE
range("B4") = QE
range("F28") = calc_range
range("B3") = des_v0


End Sub
```

APPENDIX E. VISUAL BASIC CODE (QE FINDER TRAJECTORY)

```vb
Option Explicit

Dim initialvelocity, QEradians, launchalt, tgtalt
Dim vx_AG, vz_AG
Dim frho, fT, fl
Dim g, m, d, Ixx, twist, s, k, R, tgt_ht, dt, i As Integer
Dim initialvx, initialvy, initialp

Dim projrange, impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord,
maxordrange, TOF

Sub QEFinderTrajectory(QE, calc_range)

'Getting Firing Inputs
initialvelocity = range("B28")
QEradians = QE / 3200 * WorksheetFunction.Pi
launchalt = range("B5")
tgtalt = range("B6")

'Getting Weapon Data
twist = Worksheets("Aerodata").range("B25")
Ixx = Worksheets("Aerodata").range("B26")
m = Worksheets("Aerodata").range("B27")
d = Worksheets("Aerodata").range("B28")

'Getting Wind Data
vx_AG = range("B15")
vz_AG = range("B16")

'Getting Form Factors
fl = Worksheets("Aerodata").range("B31")
fT = Worksheets("Aerodata").range("B32")
frho = Worksheets("Aerodata").range("B33")

'Getting General Parameters
dt = range("B9")
g = range("B10")
k = range("B11").Value / range("B12").Value
R = range("B11").Value - range("B12").Value
i = 0

'Initial Velocities
```

```
initialvx = initialvelocity * Cos(QEradians)
initialvy = initialvelocity * Sin(QEradians)
initialp = 2 * WorksheetFunction.Pi * initialvelocity / (twist * d)

'Reference Area
s = 0.25 * WorksheetFunction.Pi * d ^ 2

Call TrajectoryComputer(initialvelocity, QEradians, launchalt, tgtalt, twist, Ixx, m, d,
vx_AG, vz_AG, fl, fT, frho, dt, g, k, R, i, initialvx, initialvy, initialp, s, projrange,
impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange, TOF)

' Output Results
calc_range = projrange

End Sub
```

APPENDIX F. VISUAL BASIC CODE (NOMINAL TRAJECTORY)

```vb
Option Explicit

Dim initialvelocity, QEradians, launchalt, tgtalt
Dim vx_AG, vz_AG
Dim frho, fT, fl
Dim g, m, d, Ixx, twist, s, k, R, tgt_ht, dt, i As Integer
Dim initialvx, initialvy, initialp

Dim projrange, impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord,
maxordrange, TOF
Sub NominalTrajectory()

'Computation indicator switch to computing
range("E19").Interior.ColorIndex = 3
range("F19").Interior.ColorIndex = 2

'Clearing output cells
range("F2:F11").Select
Selection.clearcontents

'Getting Firing Inputs
initialvelocity = range("B3")
QEradians = range("B4") * WorksheetFunction.Pi / 3200
launchalt = range("B5")
tgtalt = range("B6")

'Getting Weapon Data
twist = Worksheets("Aerodata").range("B25")
Ixx = Worksheets("Aerodata").range("B26")
m = Worksheets("Aerodata").range("B27")
d = Worksheets("Aerodata").range("B28")

'Getting Wind Data
vx_AG = range("B15")
vz_AG = range("B16")

'Getting Form Factors
fl = Worksheets("Aerodata").range("B31")
fT = Worksheets("Aerodata").range("B32")
frho = Worksheets("Aerodata").range("B33")

'Getting General Parameters
```

```
dt = range("B9")
g = range("B10")
k = range("B11").Value / range("B12").Value
R = range("B11").Value - range("B12").Value
i = 0

'Initial Velocities
initialvx = initialvelocity * Cos(QEradians)
initialvy = initialvelocity * Sin(QEradians)
initialp = 2 * WorksheetFunction.Pi * initialvelocity / (twist * d)

'Ref Area
s = 0.25 * WorksheetFunction.Pi * d ^ 2

Call TrajectoryComputer(initialvelocity, QEradians, launchalt, tgtalt, twist, Ixx, m, d,
vx_AG, vz_AG, fl, fT, frho, dt, g, k, R, i, initialvx, initialvy, initialp, s, projrange,
impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange, TOF)

'Writing Outputs

range("F2") = QEradians * 3200 / WorksheetFunction.Pi
range("F3") = projrange
range("F4") = initialvelocity
range("F5") = impvelocity
range("F6") = impangle
range("F7") = totaldrift
range("F8") = totaldriftmils
range("F9") = maxord
range("F10") = maxordrange
range("F11") = TOF

'Computation indicator switch to computation complete
range("E19").Interior.ColorIndex = 2
range("F19").Interior.ColorIndex = 35

End Sub
```

APPENDIX G. VISUAL BASIC CODE (PARTIALS MAIN)

```
Sub PartialsMain()

'Computation indicator switch to computing
range("E19").Interior.ColorIndex = 3
range("F19").Interior.ColorIndex = 2

If range("B7") = "Predicted" Then
    PrecisionError
    MPIErrorPredicted
End If

If range("B7") = "Adjusted" Then
    PrecisionError
    'MPIErrorAdjusted (not developed)
End If

'Computation indicator switch to computation complete
range("E19").Interior.ColorIndex = 2
range("F19").Interior.ColorIndex = 35

End Sub
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX H. VISUAL BASIC CODE (PARTIALS MPI ERROR PREDICTED)

Option Explicit

```
Dim initialvelocity, QEradians, launchalt, tgtalt
Dim vx_AG, vz_AG
Dim frho, fT, fl
Dim g, m, d, Ixx, twist, s, k, R, tgt_ht, dt, i As Integer
Dim initialvx, initialvy, initialp

Dim projrange, impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord,
maxordrange, TOF
Sub MPIErrorPredicted()

'*****************************************************************************
'Perturbation in Tailwind
'*****************************************************************************

'Getting Firing Inputs
initialvelocity = range("B3")
QEradians = range("B4") * WorksheetFunction.Pi / 3200
launchalt = range("B5")
tgtalt = range("B6")

'Getting Weapon Data
twist = Worksheets("Aerodata").range("B25")
Ixx = Worksheets("Aerodata").range("B26")
m = Worksheets("Aerodata").range("B27")
d = Worksheets("Aerodata").range("B28")

'Getting Wind Data
vx_AG = range("B15") + range("J29").Value * 0.514444
vz_AG = range("B16")

'Getting Form Factors
fl = Worksheets("Aerodata").range("B31")
fT = Worksheets("Aerodata").range("B32")
frho = Worksheets("Aerodata").range("B33")

'Getting General Parameters
dt = range("B9")
g = range("B10")
k = range("B11").Value / range("B12").Value
```

```
R = range("B11").Value - range("B12").Value
i = 0

'Initial Velocities
initialvx = initialvelocity * Cos(QEradians)
initialvy = initialvelocity * Sin(QEradians)
initialp = 2 * WorksheetFunction.Pi * initialvelocity / (twist * d)

'Ref Area
s = 0.25 * WorksheetFunction.Pi * d ^ 2

Call TrajectoryComputer(initialvelocity, QEradians, launchalt, tgtalt, twist, Ixx, m, d,
vx_AG, vz_AG, fl, fT, frho, dt, g, k, R, i, initialvx, initialvy, initialp, s, projrange,
impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange, TOF)

' Writing Results
With Worksheets("MPI Error")
    .range("D10") = QEradians / WorksheetFunction.Pi * 3200
    .range("E10") = projrange
    .range("F10") = initialvelocity
    .range("G10") = impvelocity
    .range("H10") = impangle
    .range("I10") = totaldrift
    .range("J10") = totaldriftmils
    .range("K10") = maxord
    .range("L10") = maxordrange
    .range("M10") = TOF
End With

'*********************************************************************
'Perturbation in Crosswind
'*********************************************************************

'Getting Firing Inputs
initialvelocity = range("B3")
QEradians = range("B4") * WorksheetFunction.Pi / 3200
launchalt = range("B5")
tgtalt = range("B6")

'Getting Weapon Data
twist = Worksheets("Aerodata").range("B25")
Ixx = Worksheets("Aerodata").range("B26")
m = Worksheets("Aerodata").range("B27")
d = Worksheets("Aerodata").range("B28")
```

```vba
'Getting Wind Data
vx_AG = range("B15")
vz_AG = range("B16") + range("J30").Value * 0.514444

'Getting Form Factors
fl = Worksheets("Aerodata").range("B31")
fT = Worksheets("Aerodata").range("B32")
frho = Worksheets("Aerodata").range("B33")

'Getting General Parameters
dt = range("B9")
g = range("B10")
k = range("B11").Value / range("B12").Value
R = range("B11").Value - range("B12").Value
i = 0

'Initial Velocities
initialvx = initialvelocity * Cos(QEradians)
initialvy = initialvelocity * Sin(QEradians)
initialp = 2 * WorksheetFunction.Pi * initialvelocity / (twist * d)

'Ref Area
s = 0.25 * WorksheetFunction.Pi * d ^ 2

Call TrajectoryComputer(initialvelocity, QEradians, launchalt, tgtalt, twist, Ixx, m, d, _
vx_AG, vz_AG, fl, fT, frho, dt, g, k, R, i, initialvx, initialvy, initialp, s, projrange, _
impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange, TOF)

' Writing Results
With Worksheets("MPI Error")
    .range("D12") = QEradians / WorksheetFunction.Pi * 3200
    .range("E12") = projrange
    .range("F12") = initialvelocity
    .range("G12") = impvelocity
    .range("H12") = impangle
    .range("I12") = totaldrift
    .range("J12") = totaldriftmils
    .range("K12") = maxord
    .range("L12") = maxordrange
    .range("M12") = TOF
End With

'******************************************************************************
'Perturbation in air temperature
'******************************************************************************
```

```vba
'Getting Firing Inputs
initialvelocity = range("B3")
QEradians = range("B4") * WorksheetFunction.Pi / 3200
launchalt = range("B5")
tgtalt = range("B6")

'Getting Weapon Data
twist = Worksheets("Aerodata").range("B25")
Ixx = Worksheets("Aerodata").range("B26")
m = Worksheets("Aerodata").range("B27")
d = Worksheets("Aerodata").range("B28")

'Getting Wind Data
vx_AG = range("B15")
vz_AG = range("B16")

'Getting Form Factors
fl = Worksheets("Aerodata").range("B31")
fT = (1 + range("J31").Value / 100) * Worksheets("Aerodata").range("B32")
frho = Worksheets("Aerodata").range("B33")

'Getting General Parameters
dt = range("B9")
g = range("B10")
k = range("B11").Value / range("B12").Value
R = range("B11").Value - range("B12").Value
i = 0

'Initial Velocities
initialvx = initialvelocity * Cos(QEradians)
initialvy = initialvelocity * Sin(QEradians)
initialp = 2 * WorksheetFunction.Pi * initialvelocity / (twist * d)

'Ref Area
s = 0.25 * WorksheetFunction.Pi * d ^ 2

Call TrajectoryComputer(initialvelocity, QEradians, launchalt, tgtalt, twist, Ixx, m, d, _
vx_AG, vz_AG, fl, fT, frho, dt, g, k, R, i, initialvx, initialvy, initialp, s, projrange, _
impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange, TOF)

' Writing Results
With Worksheets("MPI Error")
    .range("D14") = QEradians / WorksheetFunction.Pi * 3200
    .range("E14") = projrange
```

```
        .range("F14") = initialvelocity
        .range("G14") = impvelocity
        .range("H14") = impangle
        .range("I14") = totaldrift
        .range("J14") = totaldriftmils
        .range("K14") = maxord
        .range("L14") = maxordrange
        .range("M14") = TOF
End With

End Sub
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I. VISUAL BASIC CODE (PARTIALS PRECISION ERROR)

```
Option Explicit

Dim initialvelocity, QEradians, launchalt, tgtalt
Dim vx_AG, vz_AG
Dim frho, fT, fl
Dim g, m, d, Ixx, twist, s, k, R, tgt_ht, dt, i As Integer
Dim initialvx, initialvy, initialp

Dim projrange, impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord,
maxordrange, TOF
Sub PrecisionError()

'*************************************************************************
'Perturbation in Initial Velocity
'*************************************************************************

'Getting Firing Inputs
initialvelocity = range("B3") + range("J34")
QEradians = range("B4") * WorksheetFunction.Pi / 3200
launchalt = range("B5")
tgtalt = range("B6")

'Getting Weapon Data
twist = Worksheets("Aerodata").range("B25")
Ixx = Worksheets("Aerodata").range("B26")
m = Worksheets("Aerodata").range("B27")
d = Worksheets("Aerodata").range("B28")

'Getting Wind Data
vx_AG = range("B15")
vz_AG = range("B16")

'Getting Form Factors
fl = Worksheets("Aerodata").range("B31")
fT = Worksheets("Aerodata").range("B32")
frho = Worksheets("Aerodata").range("B33")

'Getting General Parameters
dt = range("B9")
g = range("B10")
k = range("B11").Value / range("B12").Value
```

```
R = range("B11").Value - range("B12").Value
i = 0

'Initial Velocities
initialvx = initialvelocity * Cos(QEradians)
initialvy = initialvelocity * Sin(QEradians)
initialp = 2 * WorksheetFunction.Pi * initialvelocity / (twist * d)

'Ref Area
s = 0.25 * WorksheetFunction.Pi * d ^ 2

Call TrajectoryComputer(initialvelocity, QEradians, launchalt, tgtalt, twist, Ixx, m, d,
vx_AG, vz_AG, fl, fT, frho, dt, g, k, R, i, initialvx, initialvy, initialp, s, projrange,
impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange, TOF)

' Writing Results
With Worksheets("Precision Error")
    .range("E11") = QEradians / WorksheetFunction.Pi * 3200
    .range("F11") = projrange
    .range("G11") = initialvelocity
    .range("H11") = impvelocity
    .range("I11") = impangle
    .range("J11") = totaldrift
    .range("K11") = totaldriftmils
    .range("L11") = maxord
    .range("M11") = maxordrange
    .range("N11") = TOF
End With


'*********************************************************************
'Perturbation in air density
'*********************************************************************

'Getting Firing Inputs
initialvelocity = range("B3")
QEradians = range("B4") * WorksheetFunction.Pi / 3200
launchalt = range("B5")
tgtalt = range("B6")

'Getting Weapon Data
twist = Worksheets("Aerodata").range("B25")
Ixx = Worksheets("Aerodata").range("B26")
m = Worksheets("Aerodata").range("B27")
d = Worksheets("Aerodata").range("B28")
```

```vba
'Getting Wind Data
vx_AG = range("B15")
vz_AG = range("B16")

'Getting Form Factors
fl = Worksheets("Aerodata").range("B31")
fT = Worksheets("Aerodata").range("B32")
frho = (1 + range("J32").Value / 100) * Worksheets("Aerodata").range("B33")

'Getting General Parameters
dt = range("B9")
g = range("B10")
k = range("B11").Value / range("B12").Value
R = range("B11").Value - range("B12").Value
i = 0

'Initial Velocities
initialvx = initialvelocity * Cos(QEradians)
initialvy = initialvelocity * Sin(QEradians)
initialp = 2 * WorksheetFunction.Pi * initialvelocity / (twist * d)

'Ref Area
s = 0.25 * WorksheetFunction.Pi * d ^ 2

Call TrajectoryComputer(initialvelocity, QEradians, launchalt, tgtalt, twist, Ixx, m, d,
vx_AG, vz_AG, fl, fT, frho, dt, g, k, R, i, initialvx, initialvy, initialp, s, projrange,
impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange, TOF)

' Writing Results
With Worksheets("Precision Error")
    .range("E13") = QEradians / WorksheetFunction.Pi * 3200
    .range("F13") = projrange
    .range("G13") = initialvelocity
    .range("H13") = impvelocity
    .range("I13") = impangle
    .range("J13") = totaldrift
    .range("K13") = totaldriftmils
    .range("L13") = maxord
    .range("M13") = maxordrange
    .range("N13") = TOF
End With


'****************************************************************************
```

```vba
'Perturbation in QE
'*************************************************************************

'Getting Firing Inputs
initialvelocity = range("B3")
QEradians = (range("B4") + range("J33")) * WorksheetFunction.Pi / 3200
launchalt = range("B5")
tgtalt = range("B6")

'Getting Weapon Data
twist = Worksheets("Aerodata").range("B25")
Ixx = Worksheets("Aerodata").range("B26")
m = Worksheets("Aerodata").range("B27")
d = Worksheets("Aerodata").range("B28")

'Getting Wind Data
vx_AG = range("B15")
vz_AG = range("B16")

'Getting Form Factors
fl = Worksheets("Aerodata").range("B31")
fT = Worksheets("Aerodata").range("B32")
frho = Worksheets("Aerodata").range("B33")

'Getting General Parameters
dt = range("B9")
g = range("B10")
k = range("B11").Value / range("B12").Value
R = range("B11").Value - range("B12").Value
i = 0

'Initial Velocities
initialvx = initialvelocity * Cos(QEradians)
initialvy = initialvelocity * Sin(QEradians)
initialp = 2 * WorksheetFunction.Pi * initialvelocity / (twist * d)

'Ref Area
s = 0.25 * WorksheetFunction.Pi * d ^ 2

Call TrajectoryComputer(initialvelocity, QEradians, launchalt, tgtalt, twist, Ixx, m, d,
vx_AG, vz_AG, fl, fT, frho, dt, g, k, R, i, initialvx, initialvy, initialp, s, projrange,
impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange, TOF)

' Writing Results
With Worksheets("Precision Error")
```

```
        .range("E15") = QEradians / WorksheetFunction.Pi * 3200
        .range("F15") = projrange
        .range("G15") = initialvelocity
        .range("H15") = impvelocity
        .range("I15") = impangle
        .range("J15") = totaldrift
        .range("K15") = totaldriftmils
        .range("L15") = maxord
        .range("M15") = maxordrange
        .range("N15") = TOF
End With

End Sub
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX J. VISUAL BASIC CODE (TRAJECTORY COMPUTER)

```
Option Explicit

Dim x(10000), y(10000), z(10000), p(10000), vx(10000), vy(10000), vz(10000),
v(10000)
Dim vx_PA(10000), vz_PA(10000), v_PA(10000), vm(10000)
Dim vx_AG, vz_AG
Dim ax(10000), ay(10000), az(10000), ap(10000), rolldamp(10000)
Dim theta(10000), beta(10000), phi(10000), mach(10000)

Dim frho, fT, fl

Dim Cd_a0(10000), Cd_a1(10000), Cd(10000)
Dim CMa_a0(10000), CMa_a1(10000), CMa(10000)
Dim CLa_a0(10000), CLa_a1(10000), CLa(10000)
Dim Clp_a0(10000), Clp_a1(10000), Clp(10000)
Dim Temp_a0(10000), Temp_a1(10000), Temp(10000)
Dim rho_a0(10000), rho_a1(10000), rho(10000)

Dim g, m, d, Ixx, twist, s, k, R, t(10000), dt, i As Integer

Dim Fd(10000), Fdx(10000), Fdy(10000), Fdz(10000), Flz(10000)

Dim projrange, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange

Sub TrajectoryComputer(initialvelocity, QEradians, launchalt, tgtalt, twist, Ixx, m, d,
vx_AG, vz_AG, fl, fT, frho, dt, g, k, R, i, initialvx, initialvy, initialp, s, projrange,
impvelocity, impangle, winddrift, totaldrift, totaldriftmils, maxord, maxordrange, TOF)

v(0) = initialvelocity
theta(0) = QEradians
y(0) = launchalt
vx(0) = initialvx
vy(0) = initialvy
p(0) = initialp

    While y(i) >= tgtalt Or x(i) < 500
        i = i + 1
        t(i) = t(i - 1) + dt

        'Air Properties Interpolation
```

Temp_a0(i - 1) =
WorksheetFunction.Index(Worksheets("Airtable").range("C1:C23"),
WorksheetFunction.Match(y(i - 1), Worksheets("Airtable").range("A1:A23")) + 1)
Temp_a1(i - 1) =
WorksheetFunction.Index(Worksheets("Airtable").range("D1:D23"),
WorksheetFunction.Match(y(i - 1), Worksheets("Airtable").range("A1:A23")) + 1)
Temp(i - 1) = fT * (Temp_a0(i - 1) + Temp_a1(i - 1) * y(i - 1))

rho_a0(i - 1) =
WorksheetFunction.Index(Worksheets("Airtable").range("F1:F23"),
WorksheetFunction.Match(y(i - 1), Worksheets("Airtable").range("A1:A23")) + 1)
rho_a1(i - 1) =
WorksheetFunction.Index(Worksheets("Airtable").range("G1:G23"),
WorksheetFunction.Match(y(i - 1), Worksheets("Airtable").range("A1:A23")) + 1)
rho(i - 1) = frho * (rho_a0(i - 1) + rho_a1(i - 1) * y(i - 1))

vm(i - 1) = Sqr(k * R * Temp(i - 1))   'speed of sound
vx_PA(i - 1) = vx(i - 1) - vx_AG   'x-component
relative velocity
vz_PA(i - 1) = vz(i - 1) - vz_AG   'z-component
relative velocity
v_PA(i - 1) = Sqr(vx_PA(i - 1) ^ 2 + vy(i - 1) ^ 2 + vz_PA(i - 1))   'total relative
velocity
mach(i - 1) = v_PA(i - 1) / vm(i - 1)   'relative mach no

'Aero Coefficients Interpolation
Cd_a0(i - 1) =
WorksheetFunction.Index(Worksheets("Aerodata").range("C2:C13"),
WorksheetFunction.Match(mach(i - 1), Worksheets("Aerodata").range("B2:B13")) + 1)
Cd_a1(i - 1) =
WorksheetFunction.Index(Worksheets("Aerodata").range("D2:D13"),
WorksheetFunction.Match(mach(i - 1), Worksheets("Aerodata").range("B2:B13")) + 1)
Cd(i - 1) = Cd_a0(i - 1) + Cd_a1(i - 1) * mach(i - 1)

CMa_a0(i - 1) =
WorksheetFunction.Index(Worksheets("Aerodata").range("H9:H19"),
WorksheetFunction.Match(mach(i - 1), Worksheets("Aerodata").range("G9:G19")) + 1)
CMa_a1(i - 1) =
WorksheetFunction.Index(Worksheets("Aerodata").range("I9:I19"),
WorksheetFunction.Match(mach(i - 1), Worksheets("Aerodata").range("G9:G19")) + 1)
CMa(i - 1) = CMa_a0(i - 1) + CMa_a1(i - 1) * mach(i - 1)

CLa_a0(i - 1) =
WorksheetFunction.Index(Worksheets("Aerodata").range("H2:H6"),
WorksheetFunction.Match(mach(i - 1), Worksheets("Aerodata").range("G2:G6")) + 1)

```
        CLa_a1(i - 1) =
WorksheetFunction.Index(Worksheets("Aerodata").range("I2:I6"),
WorksheetFunction.Match(mach(i - 1), Worksheets("Aerodata").range("G2:G6")) + 1)
        CLa(i - 1) = CLa_a0(i - 1) + CLa_a1(i - 1) * mach(i - 1)

        Clp_a0(i - 1) =
WorksheetFunction.Index(Worksheets("Aerodata").range("C16:C20"),
WorksheetFunction.Match(mach(i - 1), Worksheets("Aerodata").range("B16:B20")) + 1)
        Clp_a1(i - 1) =
WorksheetFunction.Index(Worksheets("Aerodata").range("D16:D20"),
WorksheetFunction.Match(mach(i - 1), Worksheets("Aerodata").range("B16:B20")) + 1)
        Clp(i - 1) = Clp_a0(i - 1) + Clp_a1(i - 1) * mach(i - 1)




        'Yaw of Repose (Side Slip Angle)
        beta(i - 1) = 8 * Ixx * p(i - 1) * g * Cos(theta(i - 1)) / (WorksheetFunction.Pi *
rho(i - 1) * d ^ 3 * CMa(i - 1) * v_PA(i - 1) ^ 3)

        'Angle between velocity vector and x-axis
        phi(i - 1) = Abs(Atn(vz(i - 1) / vx(i - 1)))

        'Forces Computation
        Fd(i - 1) = 0.5 * rho(i - 1) * v_PA(i - 1) ^ 2 * s * Cd(i - 1)
        Fdx(i - 1) = Fd(i - 1) * Cos(theta(i - 1)) * Cos(phi(i - 1))
        Fdy(i - 1) = Fd(i - 1) * Sin(theta(i - 1))
        Fdz(i - 1) = Fd(i - 1) * Cos(theta(i - 1)) * Sin(phi(i - 1))
        Flz(i - 1) = fl * 0.5 * rho(i - 1) * v_PA(i - 1) ^ 2 * s * CLa(i - 1) * Sin(beta(i - 1))
* Cos(phi(i - 1) + beta(i - 1))

        'Equations of Motion
        ax(i - 1) = -Fdx(i - 1) / m               'equation (3.1)
        ay(i - 1) = -g - Fdy(i - 1) / m           'equation (3.2)
        az(i - 1) = -(Flz(i - 1) - Fdz(i - 1)) / m              'equation (3.)

        vx(i) = vx(i - 1) + ax(i - 1) * dt        'horizontal, equation (3.14)
        vy(i) = vy(i - 1) + ay(i - 1) * dt         'vertical, equation (3.14)
        vz(i) = vz(i - 1) + az(i - 1) * dt        'lateral, equation (3.14)

        'computation of velocity and projectile angle at next time step
        v(i) = Sqr(vx(i) ^ 2 + vy(i) ^ 2 + vz(i) ^ 2)         'equation (3.15)
        x(i) = x(i - 1) + vx(i) * dt                           'equation (3.16)
        y(i) = y(i - 1) + vy(i) * dt                           'equation (3.16)
        z(i) = z(i - 1) + vz(i) * dt                           'equation (3.16)
        theta(i) = Atn(vy(i) / Sqr(vx(i) ^ 2 + vz(i) ^ 2))     'equation (3.21)
```

```vba
        'Angular deceleration and projectile spin at next time step
        rolldamp(i - 1) = 0.5 * rho(i - 1) * v_PA(i - 1) ^ 2 * s * d * (p(i - 1) * d / v_PA(i -
1)) * Clp(i - 1) 'Equation (3.7)
        ap(i - 1) = rolldamp(i - 1) / Ixx                    'equation (3.8) For a steady
torque (moment): change in angular velocity = (moment/inertia)*time
        p(i) = p(i - 1) + ap(i - 1) * dt                     'equation (3.14)
    Wend


'Computing Outputs

projrange = Sqr(x(i) ^ 2 + z(i) ^ 2)                      'Projectile Range
impvelocity = v(i)                                       'Impact Velocity
impangle = (Abs(theta(i)) / WorksheetFunction.Pi) * 3200   'Impact Angle
winddrift = vz_AG * (t(i) - projrange / v(0))             'Drift due to wind only
totaldrift = Abs(z(i)) + winddrift                      'Total drift in m
totaldriftmils = totaldrift / (projrange / 1000)        'Total drift in mils

For k = 1 To i
    If y(k) < y(k - 1) Then
        maxord = y(k - 1)                                'Max ord
        maxordrange = x(k - 1)                           'Range at Max ord
        Exit For
    End If
Next k

TOF = t(i)                                               'TOF


End Sub
```

LIST OF REFERENCES

[1]   Fundamentals of ballistics. U.S. Army Ordnance Center and School, Aberdeen Proving Ground, MD, Special Text No. 9-153, Change 1, Nov. 1964.

[2]   G. V. Bull and C. H. Murphy, *Paris Kanonen: The Paris Guns (Wilhelmgeschutze) and Project HARP*. Herford, UK: E. S. Mittler, 1988.

[3]   C. F. Horne, *Source Records of the Great War*, vol. vi. National Alumni, 1923.

[4]   M. R. Driels, *Weaponeering: Conventional Weapon System Effectiveness*, 2nd ed., Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2013.

[5]   M. R. Driels. (n.d.). Why keep delivery errors separate. [Online]. Available: http://www.weaponeering.com. Accessed Mar. 1, 2016.

[6]   H. Hunt. (n.d.). A spinning top. Cambridge University. [Online]. Available: http://www2.eng.cam.ac.uk/~hemh/gyroscopes/spinningtop.htm. Accessed Jul. 20, 2016.

[7]   C.M. Fann, "Development of an artillery accuracy model," M.S. thesis, Dept. Mech. Aero. Eng., Monterey CA, 2006.

[8]   R. L. McCoy, *Modern Exterior Ballistics*. Atglen, PA: Schiffer Publishing. 2012.

[9]   M. E. Wessam and Z. H. Chen, "Firing precision evaluation for unguided artillery projectile," in *Int. Conf. Artificial Intell. Ind. Eng.*, 2015.

[10]   Ruprecht, N. (n.d.). How dobBullets fly. Wiesbaden, Germany. [Online]. Available: http://www.nennstiel-ruprecht.de/bullfly. Accessed June 30, 2016,

[11]   M. Skande, "Numerical solution to a nonlinear external ballistics model for a direct fire control system," M.S. thesis, Dept. Mach. Design, KTH School of Ind. Eng. Manage., Stockholm, Sweden, 2014.

[12]   R. F. Lieske and M. L. Relter, "Equations of motion for a modified point mass trajectory," Ballistic Research Laboratories, Aberdeen Proving Ground, MD, 1966.

[13]   M. Khalil, H. Abdalla, and O. Kamal, "Dispersion analysis for spinning artillery projectile," *13th Int. Conf. ASAT*, Cairo, Egypt, Mar 26–28, 2009.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California